

SE-01-TS12 Техническое задание - Backend для гостевого онбординга и отложенной авторизации MoodBoss

Версия	v1.0
Дата	22.04.2026
Статус	Техническое задание. Коммерческая смета, стоимость, сроки и порядок оплаты вынесены в отдельный документ SE-01-SM12.
Основание	Договор № 25042501 от 24.04.2025, аудит юзабилити MoodBoss; необходимость снятия барьера обязательной авторизации на первом входе.
Связанные документы	SE-01-SM12 - смета и график; действующая backend-архитектура MoodBoss: auth, users, calculations, notifications, purchases.

1. Назначение документа

Настоящее техническое задание фиксирует backend-объём работ по гостевому онбордингу и отложенной авторизации MoodBoss. Документ описывает целевую модель гостевого пользователя, позднюю привязку email / Google / Apple, сохранение данных на том же user_id, функциональные и нефункциональные требования, границы ответственности и критерии приемки.

2. Цель

Необходимо изменить backend-логику онбординга MoodBoss так, чтобы пользователь мог начать пользоваться приложением без обязательной авторизации на первом входе.

Целевой результат:

- пользователь заходит в приложение без email и пароля;
- сразу заполняет данные профиля;
- backend сразу создаёт пользовательскую сущность;
- календарь и эмоциональная погода рассчитываются и сохраняются сразу;
- на первом этапе эти данные принадлежат гостевому аккаунту;
- позже в настройках профиля пользователь может привязать этот же аккаунт к email + password, Google Sign-In или Apple Sign-In;
- после привязки весь накопленный профиль, статистика, календарь, эмоциональная погода, медиа и прочие данные остаются на том же user_id.

3. Важное архитектурное решение

Не использовать Google Analytics ID как основной backend-идентификатор пользователя.

Рекомендуемый подход:

- использовать собственный стабильный анонимный идентификатор установки, который генерируется клиентским приложением;
- использовать install_id / anonymous_id как техническую основу старта guest-сессии;
- GA ID / app_instance_id / user_pseudo_id хранить только как вспомогательный аналитический идентификатор, но не как основной ключ учётной записи.

Причины:

- GA ID не предназначен для роли логина или account identity;
- GA ID может быть нестабилен и зависеть от аналитической инфраструктуры;
- привязка бизнес-данных к GA ID создаёт ненужные риски при миграциях, reinstall и изменениях SDK.

4. Целевая backend-модель

Оптимальная модель для текущей архитектуры:

- создавать обычного backend-пользователя сразу;
- помечать его как гостевой аккаунт;

- выдавать обычные JWT-токены;
- продолжать работу во всех сервисах через тот же user_id;
- позже апгрейдить этот же аккаунт до обычного аккаунта через привязку email / Google / Apple.

Такая модель позволяет не ломать текущую авторизацию по JWT, не переписывать глубоко calculations, users, notifications и другие сервисы, а также не мигрировать расчетные данные между двумя аккаунтами.

5. Целевой пользовательский flow

Сценарий	Шаги	Результат
Первый запуск приложения	Клиент генерирует или загружает стабильный anonymous install identifier. Приложение вызывает backend endpoint старта гостевой сессии. Backend создаёт guest user и возвращает access / refresh tokens.	Приложение работает с обычной авторизованной JWT-сессией без email / password.
Первичное заполнение профиля	Приложение отправляет данные профиля на backend. Backend сохраняет профиль на guest-пользователе и запускает существующий pipeline пересчёта.	Календарь и эмоциональная погода доступны сразу в рамках того же user_id.
Поздняя привязка аккаунта	В настройках профиля пользователь выбирает email + password, Google или Apple. Backend подтверждает выбранный метод и привязывает identity к текущему user_id.	Аккаунт перестаёт быть guest, история остаётся на месте.
Последующие входы	Пользователь входит через email/password, Google или Apple. Backend ищет привязанную identity.	Открывается тот же аккаунт и тот же user_id.

6. Объём backend-работ по сервисам

6.1 Сервис users

Необходимо:

- расширить UserModel для поддержки гостевых аккаунтов;
- добавить хранение анонимного идентификатора установки, например install_id_hash;
- добавить поля состояния гостевого аккаунта: is_guest, registered_at, upgraded_at, registration_source при необходимости;
- сохранить уникальность email, но для guest использовать placeholder email;
- для guest-пользователя задавать unusable password до момента привязки email/password;
- создавать push settings для guest-пользователей так же, как для обычных;
- возвращать информацию о guest-state в /users/api/v2/my;
- сохранить текущую логику сохранения профиля и запуска пересчётов.

Рекомендуется добавить отдельную таблицу социальных идентичностей UserIdentityModel:

- user_id;
- provider (google, apple);
- provider_subject;
- provider_email - опционально;
- timestamps;
- unique constraint на (provider, provider_subject).

Новые gRPC методы или эквивалентный внутренний service layer:

- CreateGuestUser;
- BindGuestToEmail;
- LinkSocialIdentity;
- AuthBySocialIdentity;
- GetUserByInstallId - опционально для идиempотентности.

6.2 Сервис auth

Необходимо:

- добавить endpoint старта гостевой сессии POST /auth/api/v1/guest/start;

- добавить flow привязки email для текущего guest-пользователя: отправка письма подтверждения, подтверждение email, установка пароля, перевод guest-аккаунта в обычный;
- добавить flow привязки Google;
- добавить flow привязки Apple;
- добавить login через Google;
- добавить login через Apple;
- сохранить текущий login по email/password для обычных пользователей;
- сохранить выпуск JWT по тому же user_id;
- убедиться, что Redis-логика хранения активных токенов совместима с guest-пользователями.

Endpoint	Назначение
POST /auth/api/v1/guest/start	Старт guest-сессии и выдача JWT.
POST /auth/api/v1/bind/email/send	Отправка письма подтверждения email.
POST /auth/api/v1/bind/email/confirm	Подтверждение email и установка пароля.
POST /auth/api/v1/bind/google	Привязка Google identity к текущему guest account.
POST /auth/api/v1/bind/apple	Привязка Apple identity к текущему guest account.
POST /auth/api/v1/google/login	Вход через Google в существующий привязанный аккаунт.
POST /auth/api/v1/apple/login	Вход через Apple в существующий привязанный аккаунт.

6.3 Сервис calculations

Ожидаются минимальные изменения. Необходимо проверить, что guest JWT работает так же, как текущий JWT, так как payload содержит обычный user_id. Обязателен regression-check по female calculations, male calculations, calendar, emotional weather, current day endpoints, metrics и media flows. Точечные правки выполняются только там, где есть неявное ожидание, что у пользователя уже есть email или non-guest state.

6.4 Прочие backend-сервисы

Для notifications, purchases и других зависимых сервисов необходимо проверить совместимость с guest-пользователями. Крупных архитектурных изменений не ожидается, если сервисы уже работают через user_id. Исправляются только найденные compatibility issues.

7. Функциональные требования

Блок	Требование
Создание гостевого аккаунта	Backend создаёт guest user один раз на новый install_id, возвращает валидные access / refresh tokens, создаёт default push settings и позволяет обращаться к защищённым профильным endpoints.
Обновление профиля guest	Backend принимает первый профиль от guest-пользователя, сохраняет профиль в существующие структуры пользователей и профилей, запускает текущую очередь пересчёта и делает расчёты доступными в рамках того же аккаунта.
Привязка email	Backend позволяет guest из авторизованной сессии отправить письмо подтверждения email, подтвердить владение email, установить пароль после подтверждения и перевести guest-account в обычный email-account. Если email уже привязан к другому аккаунту, возвращается business error, если отдельно не согласован merge.
Привязка Google / Apple	Backend валидирует provider token на backend-side, извлекает стабильный provider subject (sub), привязывает подтверждённую identity к текущему guest-account, не допускает привязку одной social identity к нескольким аккаунтам и поддерживает последующий вход в тот же аккаунт.
Коллизии с существующими аккаунтами	В первой версии автоматический merge существующих аккаунтов не выполняется. Если email / Google / Apple identity уже привязаны к другому пользователю, backend возвращает понятную business error.

8. Нефункциональные требования

- полная обратная совместимость для текущих email/password пользователей;
- безопасные production migrations;
- автоматические тесты на все новые auth flow;
- backend-side verification для Google и Apple;

- недопустимо тихое переназначение истории между разными user_id.
- отсутствие зависимости от GA ID как от основного account key.

9. Что не входит в объём ТЗ

- mobile-разработка;
- frontend / UI в настройках;
- внедрение analytics SDK;
- BI / reporting;
- движок merge двух аккаунтов;
- support tooling для ручных merge;
- переработка подписок сверх проверки guest compatibility;
- тексты, локализация и продуктовые сценарии вне минимально необходимой backend-части.

10. Критерии приёмки

1. Новый пользователь может начать гостевую сессию без email/password.
2. Guest получает валидный JWT и может вызывать существующие защищённые профильные endpoint'ы.
3. Guest может сохранить профиль и получить расчёты.
4. Расчёты остаются привязанными к тому же user_id
5. Guest может позже привязать email/password без потери истории.
6. Guest может позже привязать Google без потери истории.
7. Guest может позже привязать Apple без потери истории.
8. Дальнейший login через email / Google / Apple открывает тот же аккаунт.
9. Существующие обычные пользователи продолжают работать без регрессий.
10. Автоматический merge аккаунтов не выполняется, если он отдельно не согласован.

11. Состав backend-работ

В состав работ по настоящему ТЗ входят:

- изменения модели guest-аккаунта и миграции;
- поддержка anonymous install identifier;
- endpoint guest-start;
- flow привязки email для guest;
- flow Google bind + login;
- flow Apple bind + login;
- таблица social identity;
- изменения совместимости в /users/api/v2/my;
- проверка совместимости calculations;
- автоматические backend-тесты;
- подготовка релиза и сопровождение миграции.

12. Деление на этапы

Этап	Состав
Этап 1 - Guest + поздняя привязка email	guest-start; модель guest user; support install identifier; сохранение профиля; совместимость расчётов; email bind flow.
Этап 2 - Google + Apple bind и login	identity table, Google bind/login; Apple bind/login; collision rules и тесты.

13. Основные риски

- использование GA ID как primary account identity не рекомендуется;
- у Apple email может быть скрыт или отсутствовать, поэтому уникальным ключом должен быть sub;
- автоматический merge существующих аккаунтов - это отдельный и более рискованный scope;

- если мобильное приложение не сможет стабильно хранить install_id, guest continuity после reinstall будет ненадёжной.
- интеграция Google / Apple может уточнить оценку в зависимости от точного мобильного контракта.



14. Допущения

- в scope входит только backend;
- клиент будет передавать стабильный anonymous install identifier;
- клиент будет получать Google / Apple identity tokens и передавать их на backend для проверки;
- в первой версии не требуется merge уже существующего regular account с новым guest account;
- текущая Redis/JWT модель авторизации сохраняется.

15. Ответственные лица и зоны ответственности

Сторона	Ответственный	Зона ответственности
Исполнитель	Дильдин Виталий Станиславович	Организация backend-работ, постановка задач, контроль реализации, подготовка результата к сдаче.
Заказчик	ONERY OVERSEAS LIMITED / уполномоченные представители	Предоставление исходных данных, подтверждение требований, участие в проверке результата и приёмке.
Тестирование и приёмка	Зверев Владислав Иванович	Участие в проверке guest flow, regression-check и подтверждении готовности результата к утверждению.

Подписи сторон

Исполнитель	Заказчик
ООО Вистади	ONERY OVERSEAS LIMITED
Директор:  Дильдин В.С.	Директор:  Булициду А.
М.П.	М.П.

