Progress Report 3

Work Date / Hours Log

Date	Number of Hours	Description of Work Done
Oct 13	2	Refactor new backend module structure for Gluu Admin API, including routes, Supabase, Stripe, and environment configuration files. Initialized TypeScript and Express with Zod-based config validation.
Oct 14	3	Implemented backend endpoints for plan upgrade/downgrade (POST /admin/users/:id/plan). Added logic for creating/updating Stripe customers and subscriptions with proration handling and Supabase synchronization.
Oct 15	3.5	Developed account deactivation (POST /admin/users/:id/deactivate) and permanent deletion (DELETE /admin/users/:id) routes. Integrated Supabase Admin API and Stripe pause/delete operations with atomic data updates.
Oct 16	2.5	Added input validation with Zod, structured error handling (404/409 semantics), and console-based logging. Implemented security middleware enforcing ADMIN_API_KEY for admin-only access.
Oct 17	3.5	Configured CORS restrictions, finalized modular backend structure (supabase.ts, stripe.ts, env.ts), and performed local testing to ensure all admin endpoints worked end-to-end.
Oct 21	3	Integrated Supabase API on frontend for live user management data. Replaced mock data with real-time queries and updates to the users table using @supabase/supabase-js.
Oct 22	3	Implemented pagination, search, and profile loading in Users view. Verified secure RLS access with anon key and completed testing of account activation/deactivation features through frontend and backend.
Oct 23	1	Online meeting with Gluu CEO Zane, and a short meeting with group members after the meeting with Zane.

Description of Work Done

Over this two-week period, I focused on completing the Gluu Admin Backend API and integrating it with Supabase for real-time user management. The backend was implemented with Node.js, TypeScript, and Express, featuring a modular structure for routes, Supabase access, Stripe operations, and environment validation. I developed and tested three major administrative endpoints—manual plan change, account deactivation,

and permanent deletion—ensuring secure admin-only access through an API key and precise synchronization between Supabase and Stripe. Afterward, I integrated the frontend with Supabase to replace placeholder data, enabling live user listing, profile viewing, and account management. The system was fully tested end-to-end to confirm stable data flow and production readiness.

Repo Check-in of Implementation Completed

The files/folders I have checked into the repository so far include:

- Implementation/backend/src/ (branch dev)
- Implementation/backend/src/lib/plan.ts (branch dev)
- Implementation/backend/src/middleware/adminAuth.ts (branch dev)
- Implementation/backend/src/routes/admin.ts (branch dev)
- Implementation/frontend/src/lib/adminApi.ts (branch dev)
- Implementation/frontend/src/lib/supabaseClient.ts (branch dev)
- Implementation/frontend/src/pages/Users/UsersList.tsx (branch dev)
- Implementation/frontend/src/components/users/UsersTable.tsx (branch dev)
- Implementation/frontend/src/components/users/Pagination.tsx (branch dev)
- Implementation/frontend/src/components/users/PlanChangeModal.tsx (branch dev)
- Implementation/frontend/src/components/users/UserActionsModal.tsx (branch dev)
- Implementation/frontend/src/components/users/UserDetailDrawer.tsx (branch dev)