

Contents

1 Functional Requirements Document (FRD)	1
1.1 1. Document Control	1
1.2 2. Purpose	1
1.3 3. Scope	1
1.3.1 3.1 MVP Scope (Current Phase)	1
1.3.2 3.2 Future Phases (Post-MVP)	2
1.4 4. Stakeholders	2
1.5 5. Functional Requirements	2
1.5.1 5.1 MVP Functional Requirements (Current Phase)	2
1.5.2 5.2 Future Functional Requirements (Post-MVP)	2
1.6 6. Non-Functional Requirements	3
1.7 7. Assumptions & Dependencies	3
1.7.1 Data & Scripts	3
1.7.2 Other Assumptions	3
1.8 8. Use Cases	3
1.8.1 MVP Use Cases (Current Phase)	3
1.8.2 Future Use Cases (Post-MVP)	4
1.9 9. Business Rules	4
1.9.1 BR-1: Membership Tier Restrictions	4
1.9.2 BR-2: Session Booking Rules	4
1.9.3 BR-3: Equipment Management	4
1.9.4 BR-4: Access Control	5
1.9.5 BR-5: Audit Requirements	5
1.9.6 BR-6: Session Management	5
1.9.7 BR-7: Trainer Staffing	5
1.10 10. Success Metrics	5

1 Functional Requirements Document (FRD)

1.1 1. Document Control

- **Version:** 2.1
- **Author:** Henry Huerta
- **Date:** 2025-11-3
- **Reviewers:** Prof. Arnold Lau, T.A. Sneh Bhandari

1.2 2. Purpose

FitDB is a reliable, central system that supports the day-to-day functions of large, multiple-location gym chains. These functions include registering new members, tracking membership details, booking fitness class sessions, viewing trainer availability, member analytics for decision-making, and member check-ins.

1.3 3. Scope

1.3.1 3.1 MVP Scope (Current Phase)

In-Scope (MVP): - **User Account Creation:** Users can create accounts (username, email, password) - **Member Account Registration:** Front desk managers can create member accounts for users - **Access Card Issuance:** Front desk managers can issue access cards to members - **Audit Logging:** All operations are logged immutably (append-only audit trail) - **RBAC Infrastructure:** Role-based access control via MySQL roles (front desk manager role enabled)

Out-of-Scope (MVP): - Session management (trainer availability, session publishing) - Booking system (session bookings, cancellations) - Check-in system (member check-ins at gym locations) - Equipment management (inventory tracking, service logging) - Reporting and analytics (utilization reports, equipment demand) - Advanced member management (deactivation, strikes, bans) - Payroll, billing, messaging (email/SMS), wait-lists

Note: While the database schema supports all these features, the MVP focuses exclusively on account creation and access card issuance. See [MVP_SCOPE.md](#) for detailed MVP clarification.

1.3.2 3.2 Future Phases (Post-MVP)

The full scope of features listed below will be implemented in future phases: - Register/edit/deactivate/strike/ban members; member photo reference (BLOB) - Check-ins (id input) with status validation - Trainer availability entry (trainer) and session publication (manager) - Class session bookings/cancellations (plus members only) - Equipment inventory and per-session allocation checks - Reporting views (utilization, equipment demand)

1.4 4. Stakeholders

- **Member (regular):** view profile and check-ins; no booking
- **Member (plus):** all of the above + book/cancel sessions
- **Trainer:** maintain availability; view rosters
- **Manager:** publish sessions, manage equipment, adjust trainer assignment, does reporting
- **Front Desk:** check-ins; view-only member status (inherits subset from manager)
- **Floor Manager:** equipment counts/maintenance (inherits from manager)
- **Admin:** full access; role assignments; audit viewer

1.5 5. Functional Requirements

1.5.1 5.1 MVP Functional Requirements (Current Phase)

- **FR-MVP-1 User Account Creation:** create user account with username, email, password
 - **AC:** all required fields validated; username/email unique; password encrypted; USER row created; audit entry written
- **FR-MVP-2 Member Account Registration:** front desk manager creates member account for existing user
 - **AC:** user exists; membership plan valid; member row created with tier and status; appropriate role assigned; audit entry written
- **FR-MVP-3 Access Card Issuance:** front desk manager issues access card to member
 - **AC:** member exists; card UID unique; gym valid; ACCESS_CARD row created with “active” status; audit entry written

1.5.2 5.2 Future Functional Requirements (Post-MVP)

- **FR-1 Register Member:** create member with tier, status, optional photo (enhanced version with all capabilities)
- **FR-2 Publish Sessions** (manager from trainer availability)
 - **AC:** sessions created within date window; no conflicts; equipment sufficiency verified; audit entry written; rollback on any failure
- **FR-3 Book Session** (plus member)
 - **AC:** capacity and equipment checks enforced; unique booking per member/session; audit entry written; rollback on failure
- **FR-4 Check-In** (front desk)
 - **AC:** active status required; check-in saved with timestamp and method; audit entry written
- **FR-5 Reporting**

- **AC:** views return utilization and equipment demand; queries complete < 3s on 50k rows (dev target)

1.6 6. Non-Functional Requirements

- **Performance:**
 - load trainer availability in < 3s
 - retrieve member data in < 3s
- **Security:**
 - enforce RBAC
 - maintain append-only audit log
 - hash passwords/protect photos
- **Scalability:**
 - Must handle up to 1,000 concurrent users while maintaining transactional integrity
 - Provide EXPLAIN/ANALYZE snapshots
 - index strategy documented
 - paginate heavy lists
- **Usability:**
 - Users will retrieve and add data through a GUI including calendars, text-based entry, and clickable items

1.7 7. Assumptions & Dependencies

1.7.1 Data & Scripts

- provide `build.sql` to create schema & indexes; provide seed data for demo
- provide fixtures for forced-failure cases (overbooking, past-date booking).

1.7.2 Other Assumptions

- Schema supports multi-gym in future phases.
- AWS optional for production

1.8 8. Use Cases

1.8.1 MVP Use Cases (Current Phase)

1.8.1.1 UC-1: User Account Creation (MVP) **Actor:** End User **Precondition:** User account does not exist **Flow:** 1. User enters account information (username, email, password) 2. System validates username/email uniqueness 3. System creates USER record with encrypted password 4. System writes audit log entry **Postcondition:** User account exists and is ready for member registration **Exceptions:** Username/email already exists, invalid password

1.8.1.2 UC-2: Member Account Registration by Front Desk Manager (MVP) **Actor:** Front Desk Manager **Precondition:** User account exists (created via UC-1) **Flow:** 1. Front desk manager enters user ID or username 2. Front desk manager selects membership plan 3. Front desk manager enters home gym and other member details 4. System creates MEMBER record linked to USER and membership plan 5. System assigns appropriate role to user (member or plus_member) 6. System writes audit log entry **Postcondition:** Member account is created and ready for access card issuance **Exceptions:** User already has member account, invalid membership plan

1.8.1.3 UC-3: Access Card Issuance (MVP) **Actor:** Front Desk Manager **Precondition:** Member account exists (created via UC-2) **Flow:** 1. Front desk manager selects member (by ID or username) 2. Front desk manager enters card UID (unique identifier) 3. Front desk manager selects gym where card is issued 4. System creates ACCESS_CARD record linked to member and gym 5. System sets card status to

“active” 6. System writes audit log entry **Postcondition:** Member has an active access card for gym access
Exceptions: Card UID already exists, member already has active card for that gym

1.8.2 Future Use Cases (Post-MVP)

1.8.2.1 UC-4: Session Booking (Plus Member Only) **Actor:** Plus Member **Precondition:** Member has active “plus” membership, session is open for booking **Flow:** 1. Member views available sessions through `vw_bookable_sessions` 2. Member selects a session 3. System validates: member tier is “plus”, capacity not exceeded, equipment available 4. System creates `BOOKING` record with status “confirmed” 5. System writes audit log entry 6. Member receives booking confirmation **Postcondition:** Member has booked session, capacity reduced **Exceptions:** Not plus member, session full, past booking window, equipment unavailable

1.8.2.2 UC-5: Check-In **Actor:** Front Desk Staff / Member **Precondition:** Member has active membership and valid access card **Flow:** 1. Member presents access card or provides ID 2. System validates: membership is active, card not revoked/lost 3. System validates: trial/basic members at home gym only; plus members at any gym 4. System creates `CHECK_IN` record with timestamp 5. System writes audit log entry **Postcondition:** Member is checked in, attendance recorded **Exceptions:** Inactive membership, revoked/lost card, wrong gym for non-plus members

1.8.2.3 UC-6: Trainer Availability Management **Actor:** Trainer **Precondition:** Trainer is logged in **Flow:** 1. Trainer views current availability schedule 2. Trainer sets availability for date/period via `sp_set_availability` 3. System validates: date is in future, period is AM or PM 4. System creates/updates `TRAINER_AVAIL_DATE` record 5. System writes audit log entry **Postcondition:** Trainer availability updated **Exceptions:** Invalid date, conflict with existing sessions

1.8.2.4 UC-7: Session Publishing (Manager) **Actor:** Manager **Precondition:** Trainer has set availability **Flow:** 1. Manager views trainer availability 2. Manager selects availability block and creates class session 3. System validates: trainer available, equipment sufficient, no time conflicts 4. System creates `CLASS_SESSION` with default capacity 5. System creates `SESSION_TRAINER` assignment 6. System writes audit log entries **Postcondition:** Session is published, trainers assigned **Exceptions:** No trainer availability, equipment conflict, time conflict

1.9 9. Business Rules

1.9.1 BR-1: Membership Tier Restrictions

- **Trial/Basic:** Can check in only at home gym; cannot book sessions
- **Plus:** Can check in at any gym; can book sessions up to 2 months in advance
- **Member status** must be “active” for any gym access

1.9.2 BR-2: Session Booking Rules

- **Booking window:** Plus members can book sessions up to 2 months in advance
- **Capacity limit:** Cannot exceed `CLASS_SESSION.capacity`
- **Booking exclusivity:** Member can only have one confirmed booking per session
- **Equipment allocation:** System must verify sufficient equipment for all attendees

1.9.3 BR-3: Equipment Management

- Equipment items have status: OK, NEEDS_SERVICE, OUT_OF_ORDER, RETIRED
- Status cannot be “OK” if `service_required` or `cleaning_required` flags are true
- Service logs must be maintained for audit and compliance

1.9.4 BR-4: Access Control

- Access cards must be “active” for check-in
- Lost or revoked cards cannot be used for check-in
- Check-ins require valid active membership

1.9.5 BR-5: Audit Requirements

- All INSERT, UPDATE, DELETE operations must write to audit tables
- Audit records are append-only (immutable)
- Audit records include: action type, JSON snapshot, timestamp, actor
- No direct modification of audit records allowed

1.9.6 BR-6: Session Management

- Sessions can be: SCHEDULED, CANCELED, COMPLETED
- Canceled sessions release capacity and equipment
- Members with bookings for canceled sessions receive notification

1.9.7 BR-7: Trainer Staffing

- Each session can have multiple trainers (lead and assistant roles)
- Maximum trainers per session is `CLASS_SESSION.max_trainers`
- Trainers must have availability set for session date/period

1.10 10. Success Metrics

- Demoable booking with audit proof
- Check rubric