SPRINT 2

Web Technologies – QuickFix
Team: StackSquad

MAJOR UPDATES FROM SPRINT 1

Sprint 2 focused on expanding the backend infrastructure, integrating the database into the system, improving architecture documentation, and ensuring the core platform runs on a well-structured MySQL backend. The foundation created in Sprint 1 enabled the team to implement more robust features in this sprint.

The team successfully:

- Designed a full **database ER diagram**
- Implemented a complete **database schema** based on the ERD
- Integrated MySQL into all major pages (registration, login, browsing, booking)
- Developed backend queries and data connections
- Updated the system architecture to reflect actual technologies
- Delivered an expanded functional demo with live data

SPRINT 2 DELIVERABLES

1. Updated Architecture Page

The architecture was revised to reflect:

- **Three-tier architecture** (Presentation → Logic → Database)
- MySQL integration into all dynamic pages

- Backend modules:
    - Authentication module
    - Booking engine
    - Provider management
    - Customer account system
    - Notifications and messaging stubs
- JSON-based temporary data fully replaced with **live SQL database**

The new architecture ensures clean separation, scalability, and maintainability.

2. Database Entity Relationship Diagram (ERD)

A complete ERD was designed to support the QuickFix system.
Major entities include:

- **Users**
- **Providers**
- **Services**
- **Categories**
- **Bookings**
- **Reviews**
- **Messages**
- **Payments**

- **Favorites**
- **Notifications**

Key Relationship Types

- Users ↔ Providers (1–1)
- Users ↔ Bookings (1–many)
- Services ↔ Bookings (1–many)
- Providers ↔ Services (1–many)
- Users ↔ Reviews (1–many)
- Bookings ↔ Reviews (1–1)
- Users ↔ Messages (sender/receiver)
- Providers ↔ Categories (many–1)

Attributes (Examples)

Users

- user_id (PK)
- full_name
- email
- password_hash
- phone
- role (customer/provider)
- created_at

Bookings

- booking_id (PK)
- service_id (FK)
- customer_id (FK)
- provider_id (FK)
- date
- time
- status
- created_at

3. Database Creation & SQL Schema

A full MySQL schema was implemented under your required group database name:

Database Name:

SS2027 (StackSquad 2027)

Implementation Achievements

- All tables created with primary keys, foreign keys, constraints
- Referential integrity enforced
- Correct creation order used (parent → child tables)
- Test data inserted for demo
- Connection established through `db_connection.php`
- Queries integrated into:
    - Registration

- Login
- Browse services
- Provider dashboard
- Customer dashboard
- Booking workflow

4. Backend Integration & Functional Demo

Database-Driven Features

- Registration storing user accounts in the database
- Login using real authentication queries
- Browsing services from dynamic SQL data
- Booking services with live insertion into `bookings`
- Provider dashboard loading bookings from SQL
- Customer dashboard showing booking history
- Review system reading/writing data
- Notification system triggered on booking events

New Additions

Automatic notification insert via:

```
require_once "add_notification.php";
notifyUser($provider, "You have a new booking request.");
notifyUser($customer, "Your booking was successfully submitted.");
```

-

- Database-driven dynamic content across all pages

Sprint 2 User Guide

Database-Driven Workflows

Users can now:

- Register → stored in database
- Log in → session authenticated from SQL
- Browse providers → filtered from SQL queries
- Book services → booking entry stored & notification generated
- View dashboards → dynamically loaded from SQL tables
- See reviews and ratings pulled from the database

Provider Capabilities

- Add/edit services stored in `services` table
- View bookings from customers
- Manage availability and service listings

TEAM MEMBER CONTRIBUTIONS

Lidwan / Fynn / Yelsom

- Designed and finalized ERD

- Implemented SQL schema and constraints
- Connected frontend pages to MySQL database
- Revised architecture documentation

Fynn / Derrick

- Built all backend database connections
- Wrote CRUD queries for users, providers, bookings, and services
- Implemented booking + notifications backend
- Created demo database with sample data

Yelsom / Lidwan / Derrick

- Updated dashboards to load dynamic SQL content
- Debugged database errors and query issues
- Ensured consistency with CSS and UI structure

TESTING STRATEGY

Backend Testing

- Verified all SQL tables and foreign keys
- Tested insertion/deletion edge cases
- Booking workflow tested end-to-end
- Session handling with real data verified

Frontend Testing

- All pages tested for dynamic content loading
- Form validations re-tested
- Provider/customer dashboards tested with multiple roles

Cross-Browser & Device Testing

- Chrome, Firefox, Safari
- Mobile device scaling verified

SPRINT 2 RETROSPECTIVE

What Went Well

- ER diagram completed accurately
- Database schema integrated without major blockers
- Most features now fully dynamic
- Team collaboration improved significantly
- System architecture now well-documented

What Didn’t Go Well

- Some SQL constraints required reordering tables
- Initial database connection bugs slowed early progress
- Several queries needed optimization for performance

Plans for Next Sprint (Sprint 3)

- Payment processing implementation
- Messaging system development
- Admin dashboard completion
- Notification page + real-time updates
- More automated testing
- Further database optimization