Data Validation Management System Documentation

1. Overview

A system allowing administrators to configure data validation rules for table columns through a UI interface. This replaces hardcoded validation rules with a dynamic, database-driven approach.

2. Current vs New System

Current System:

- Hardcoded validation rules in:

  - Frontend: `Number_allowed_columns.ts`, `Space_allowed_columns.ts`, `Symbol_allowed_columns.ts`

  - Backend: `numberAllowed.js`, `spaceAllowed.js`, `symbolAllowed.js`

- Fixed validation types

- Code changes required for modifications

New System:

- Database-driven validation rules

- Admin UI for configuration

- Dynamic rule application

- Instant updates without code changes

3. Database Schema

```
-- Main validation configuration table
CREATE TABLE app.column_validations (
    validation_id uuid PRIMARY KEY DEFAULT uuid_generate_v4(),
    table_name VARCHAR(100) NOT NULL,
    column_name VARCHAR(100) NOT NULL,
    -- Validation configuration
    allow_numbers BOOLEAN DEFAULT true,
```

```
    allow_special_chars BOOLEAN DEFAULT true,
    allow_spaces BOOLEAN DEFAULT true,
    -- Additional validations
    min_length INTEGER,
    max_length INTEGER,
    regex_pattern VARCHAR(500),
    custom_error_message VARCHAR(200),
    -- For numeric fields
    min_value NUMERIC,
    max_value NUMERIC,
    decimal_places INTEGER,
    -- For date fields
    min_date DATE,
    max_date DATE,
    allow_weekends BOOLEAN,
    -- Common fields
    is_active BOOLEAN DEFAULT true,
    created_at TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,
    created_by uuid REFERENCES app.users(user_id),
    UNIQUE(table_name, column_name)
);

-- Predefined validation formats
CREATE TABLE app.validation_formats (
    format_id uuid PRIMARY KEY DEFAULT uuid_generate_v4(),
    format_name VARCHAR(50) NOT NULL,
    format_pattern VARCHAR(500) NOT NULL,
    description TEXT,
    is_active BOOLEAN DEFAULT true,
    UNIQUE(format_name)
```

);

4. Validation Types

4.1 Character Validations

- Numbers (allow/disallow)
- Special characters (allow/disallow)
- Spaces (allow/disallow)
- Length restrictions

4.2 Format Validations

- Email format
- Phone number format
- Date format
- Custom regex patterns

4.3 Numeric Validations

- Range checks
- Decimal place control
- Positive/negative number control

4.4 Date Validations

- Date range control
- Weekend/holiday restrictions
- Future/past date validation

5. API Endpoints

5.1 Admin Configuration APIs

// Configuration Management

GET /api/admin/validations/tables // List all tables

GET /api/admin/validations/:tableName/columns // List columns for table

GET /api/admin/validations/rules // Get all validation rules

POST /api/admin/validations/rules // Create validation rule

PUT /api/admin/validations/rules/:validationId // Update validation rule

DELETE /api/admin/validations/rules/:validationId // Delete validation rule

PATCH /api/admin/validations/rules/:validationId/status // Toggle rule status

// Format Management

GET /api/admin/validations/formats // List validation formats

POST /api/admin/validations/formats // Add new format

PUT /api/admin/validations/formats/:formatId // Update format

DELETE /api/admin/validations/formats/:formatId // Delete format

```

**5.2 Application APIs**

GET /api/validations/:tableName // Get table validation rules

GET /api/validations/:tableName/:columnName // Get column validation rules

POST /api/validations/verify // Verify data against rules

6. Admin Interface Requirements

6.1 Table/Column Selection

- Table dropdown

- Column selection

- Data type display
```

6.2 Validation Configuration

- Basic validation toggles
- Format selection
- Range input fields
- Custom pattern input

6.3 Rule Management

- Enable/disable rules
- Bulk operations
- Rule testing interface

7. Implementation Strategy

Phase 1: Basic Setup

1. Database schema creation
2. Basic API endpoints
3. Simple validation rules

Phase 2: Enhanced Features

1. Format validations
2. Range validations
3. Custom patterns

Phase 3: UI Development

1. Admin configuration interface
2. Rule testing interface
3. Bulk operations

8. Error Handling

8.1 Configuration Errors

- Invalid validation rules
- Conflicting rules
- Missing required fields

8.2 Validation Errors

```
{
  field: string,
  value: any,
  error: string,
  validationType: string
}
```

9. Best Practices

9.1 Rule Configuration

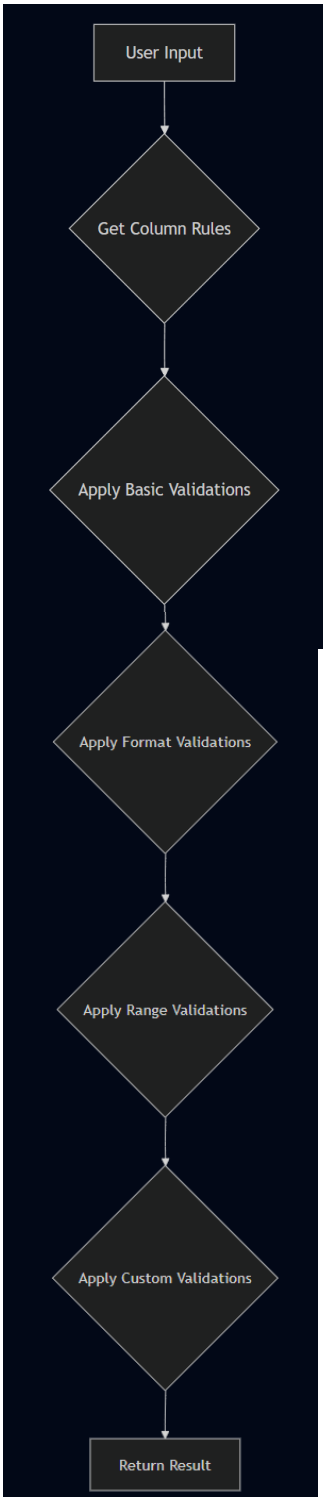
- One validation rule per column
- Clear error messages
- Consistent validation patterns

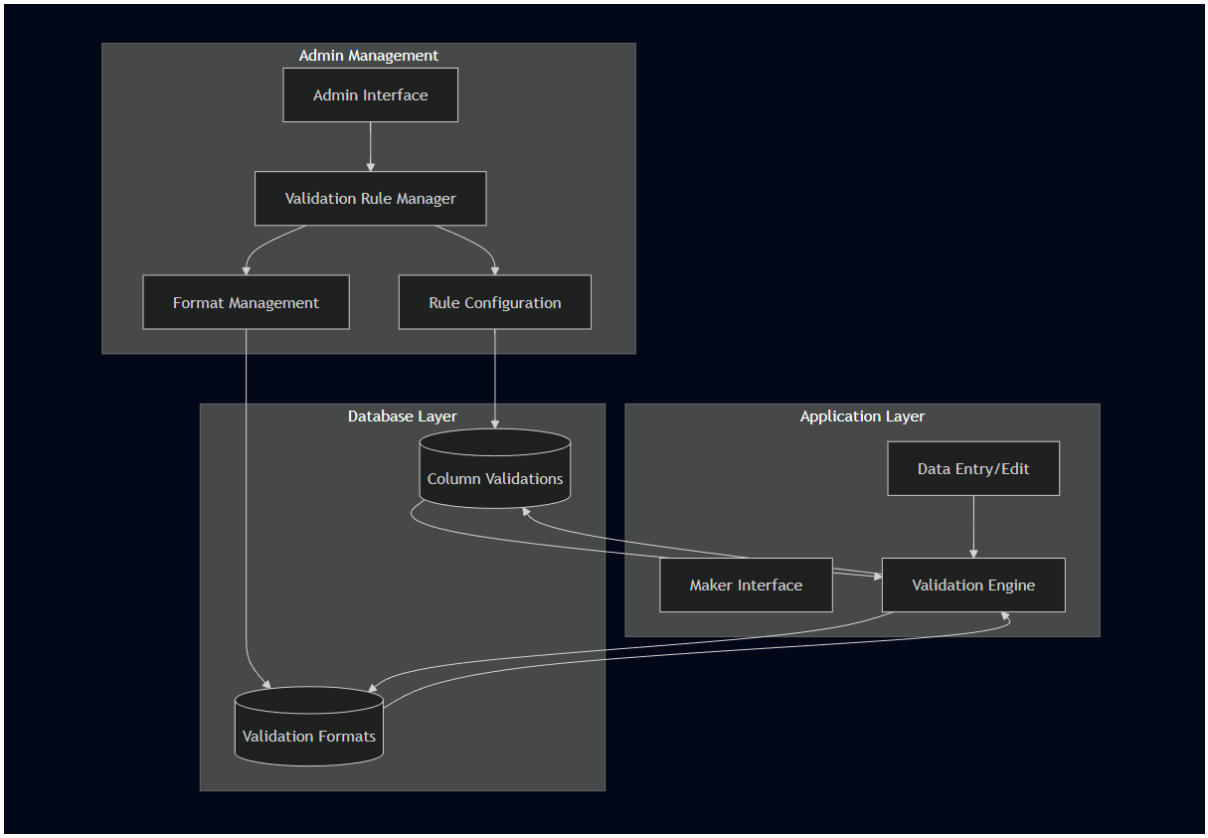
9.2 Performance

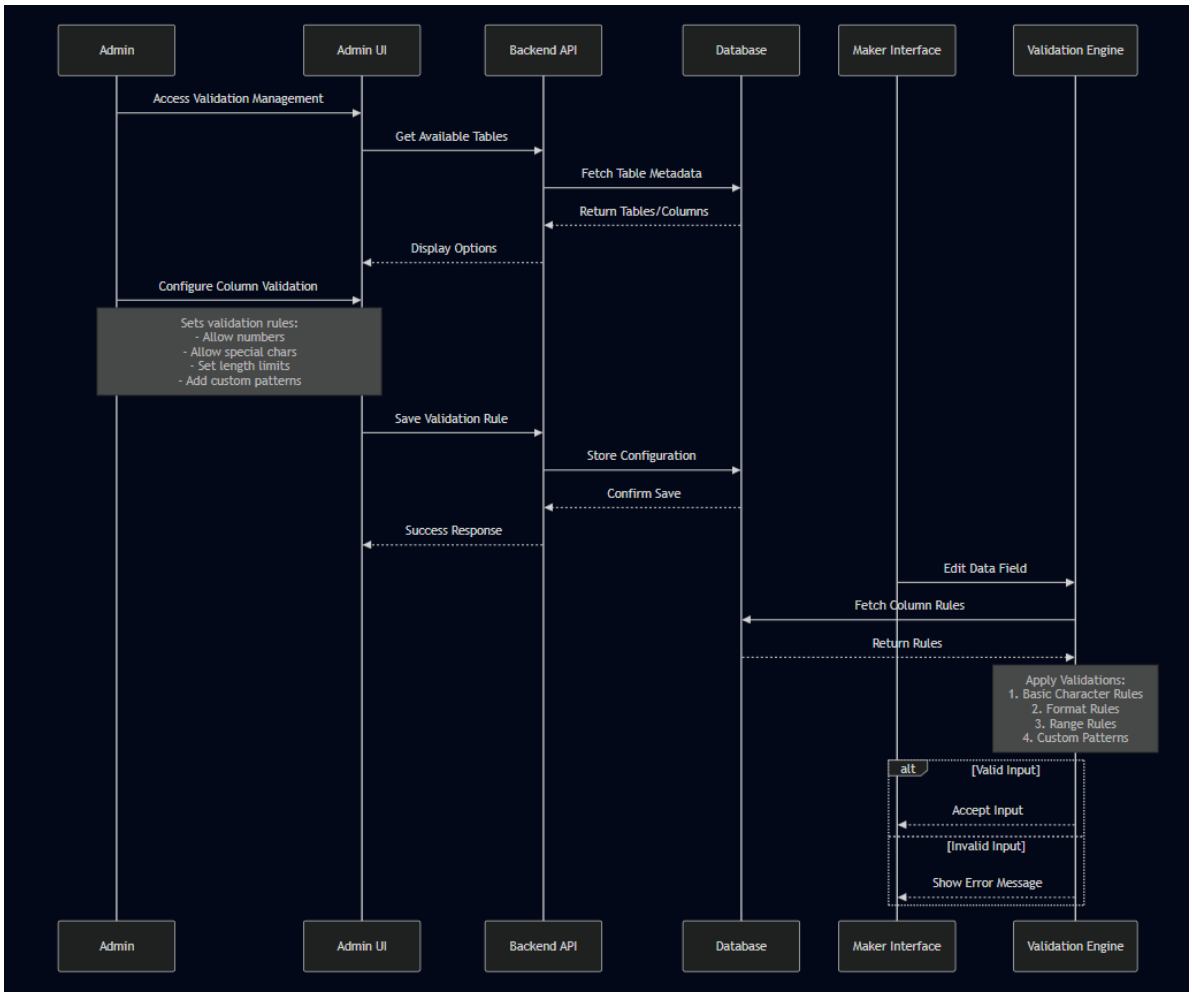
- Efficient validation execution
- Minimal database queries
- Optimized rule processing

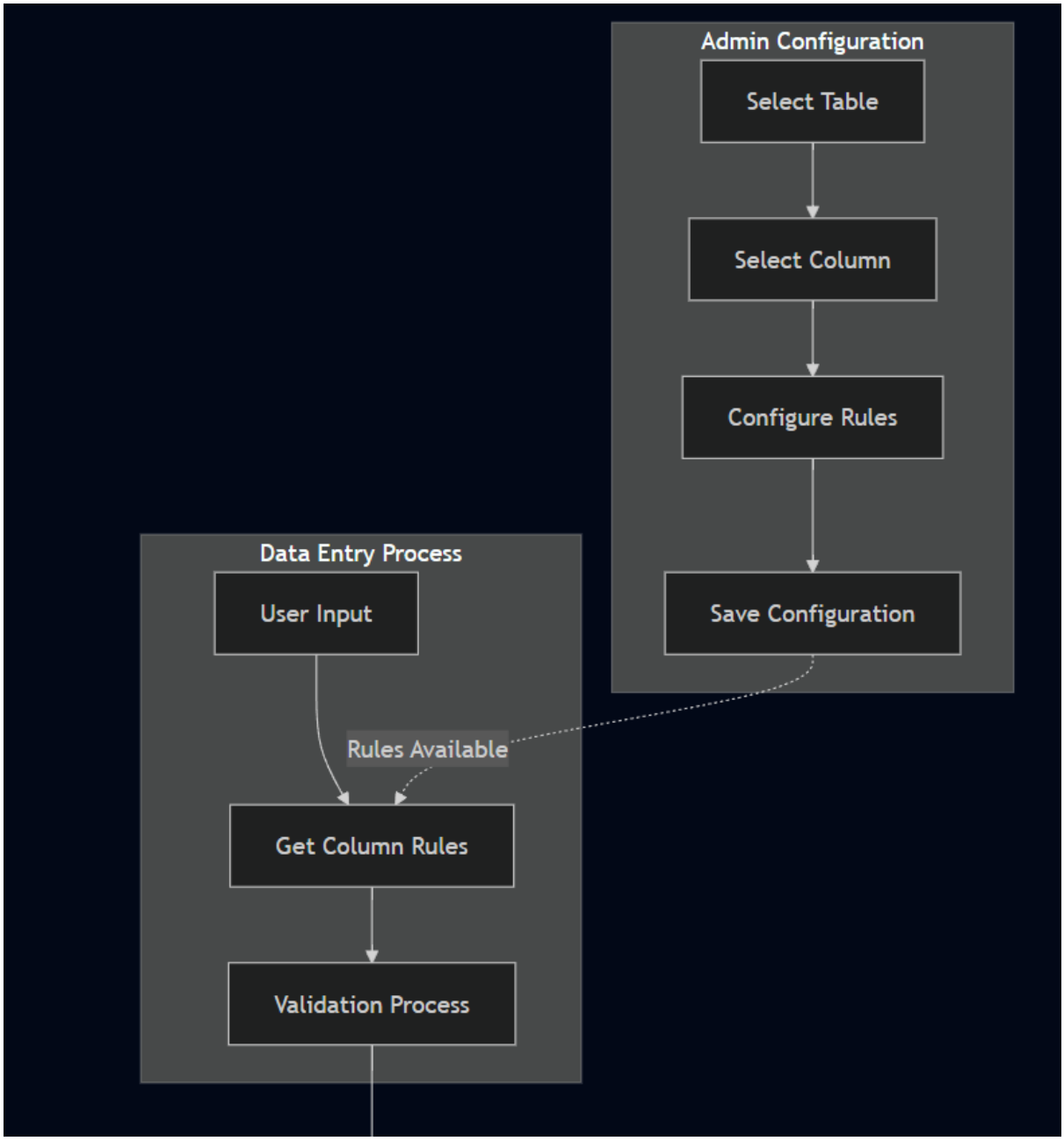
9.3 Security

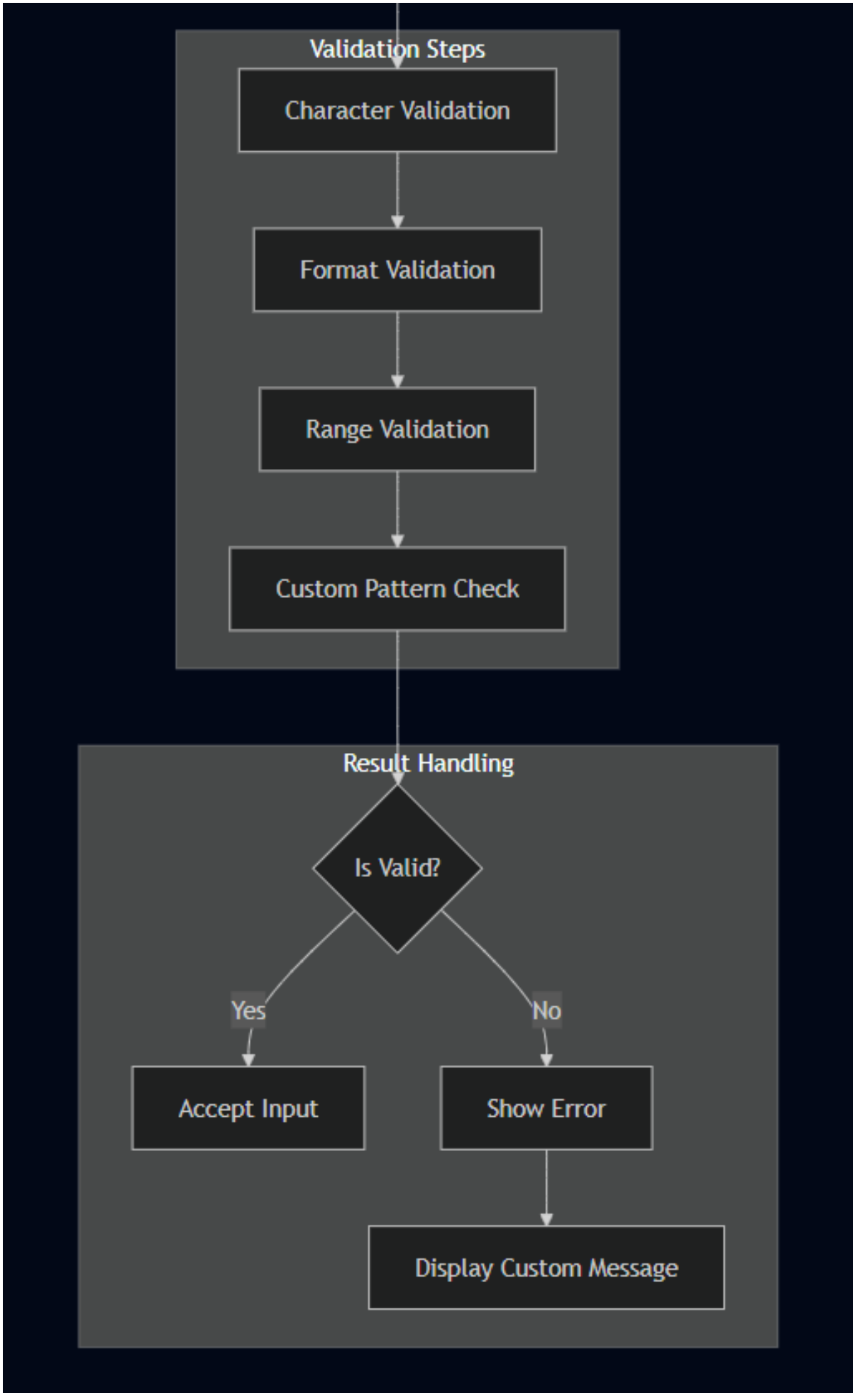
- Input sanitization
- Access control
- SQL injection prevention


User Input
Get Column Rules
Apply Basic Validations
Apply Format Validations
Apply Range Validations
Apply Custom Validations
Return Result


Admin Management
Admin Interface
Validation Rule Manager
Format Management
Rule Configuration
Database Layer
Column Validations
Validation Formats
Application Layer
Data Entry/Edit
Maker Interface
Validation Engine


Admin
Admin UI
Backend API
Database
Maker Interface
Validation Engine
Access Validation Management
Get Available Tables
Fetch Table Metadata
Return Tables/Columns
Display Options
Configure Column Validation
Sets validation rules:
- Allow numbers
- Allow special chars
- Set length limits
- Add custom patterns
Save Validation Rule
Store Configuration
Confirm Save
Success Response
Edit Data Field
Fetch Column Rules
Return Rules
Apply Validations:
1. Basic Character Rules
2. Format Rules
3. Range Rules
4. Custom Patterns
alt
[Valid Input]
Accept Input
[Invalid Input]
Show Error Message
Admin
Admin UI
Backend API
Database
Maker Interface
Validation Engine


Admin Configuration
Select Table
Select Column
Configure Rules
Save Configuration
Data Entry Process
User Input
Rules Available
Get Column Rules
Validation Process


Validation Steps
Character Validation
Format Validation
Range Validation
Custom Pattern Check
Result Handling
Is Valid?
Yes
No
Accept Input
Show Error
Display Custom Message