Product Requirements Document

Product Requirements Document: User Registration Module

1. Overview

The User Registration Module enables new users to create accounts in the application. This module provides a secure, user-friendly registration process with email verification, password strength validation, and comprehensive data validation to ensure data quality and security.

2. Business Objectives

- Enable new user account creation with minimal frictionÐ
- Ensure data quality through comprehensive validationÐ
- Implement security best practices for account creationÐ
- Support user onboarding with clear guidanceÐ
- Collect necessary user information for personalized experienceÐ
- Maintain compliance with data protection regulations (GDPR, CCPA)Ð

3. Functional Requirements

3.1 Registration Form

FR-001: Registration form must include:

- First Name field (required, max 50 characters)Ð
- Last Name field (required, max 50 characters)Ð
- Email Address field (required, max 254 characters, unique)Ð
- Password field (required, min 8 characters, masked)Ð
- Confirm Password field (required, must match password)Ð
- Phone Number field (optional, format: +1-XXX-XXX-XXXX)Ð
- Date of Birth field (optional, format: MM/DD/YYYY, must be 18+)Ð
- "I agree to Terms and Conditions" checkbox (required)Ð
- "I agree to receive marketing emails" checkbox (optional)Ð
- "Create Account" buttonÐ
- "Sign in instead" link (for existing users)Ð

FR-002: First Name validation:

- Required fieldÐ
- Minimum length: 1 characterÐ
- Maximum length: 50 charactersÐ
- Must accept alphabetic characters, spaces, hyphens, apostrophesÐ
- Must accept international characters (Unicode)Ð
- Must display error: "First name is required" if emptyÐ
- Must display error: "First name must be 50 characters or less" if too longÐ
- Must sanitize input to prevent XSS attacksÐ

FR-003: Last Name validation:

- Required fieldÐ
- Same validation rules as First NameÐ
- Must display same error messages as First NameÐ

FR-004: Email Address validation:

- Required fieldÐ
- Must validate email format (RFC 5322 compliant)Ð
- Maximum length: 254 charactersÐ
- Must check for duplicate emails in databaseÐ
- Must display error: "Email is required" if emptyÐ
- Must display error: "Please enter a valid email address" for invalid formatÐ
- Must display error: "This email is already registered. Please sign in instead." for duplicatesÐ
- Must convert to lowercase before storageÐ

FR-005: Password validation:

- Required fieldÐ
- Minimum length: 8 charactersÐ
- Must contain at least one uppercase letter (A-Z)Ð
- Must contain at least one lowercase letter (a-z)Ð
- Must contain at least one number (0-9)Ð
- Must contain at least one special character (!@#$%^&*()_+-=[]{}|;:,.<>?)Ð
- Must display real-time strength indicator (Weak/Medium/Strong)Ð
- Must display error: "Password is required" if emptyÐ
- Must display error: "Password must be at least 8 characters with uppercase, lowercase, number, and special character" for weak passwordsÐ
- Must mask password input (display as dots)Ð

FR-006: Confirm Password validation:

- Required fieldÐ
- Must exactly match Password fieldÐ
- Must display error: "Passwords do not match" if mismatchÐ
- Must validate in real-time as user typesÐ
- Must mask password input (display as dots)Ð

FR-007: Phone Number validation (optional):

- Optional fieldÐ
- Must accept format: +1-XXX-XXX-XXXX or (XXX) XXX-XXXXÐ
- Must validate phone number formatÐ
- Must display error: "Please enter a valid phone number" for invalid formatÐ
- Must store in E.164 format: +1XXXXXXXXXXÐ

FR-008: Date of Birth validation (optional):

- Optional fieldÐ
- Must be 18 years or olderÐ
- Must accept format: MM/DD/YYYYÐ
- Must validate date is not in futureÐ

- Must display error: "You must be 18 years or older to register" if under 18Đ
- Must display error: "Please enter a valid date" for invalid formatĐ

FR-009: Terms and Conditions:

- Required checkboxĐ
- Must be checked to submit formĐ
- Must display error: "You must accept the Terms and Conditions to create an account" if uncheckedĐ
- Link to Terms and Conditions page (opens in new tab)Đ

FR-010: Marketing emails:

- Optional checkboxĐ
- Default: uncheckedĐ
- If checked, user will receive marketing emailsĐ
- Preference stored in user profileĐ

3.2 Registration Processing

FR-011: Form submission:

- Validate all fields before submissionĐ
- Display loading indicator during processingĐ
- Disable submit button during processingĐ
- On validation error, highlight invalid fieldsĐ
- On success, create user account and send verification emailĐ

FR-012: Account creation:

- Create user record in database with status "pending_verification"Đ
- Hash password using bcrypt (cost factor: 12)Đ
- Generate unique verification token (UUID)Đ
- Store verification token with 24-hour expirationĐ
- Send verification email with verification linkĐ
- Log registration attempt (IP address, timestamp, user agent)Đ

FR-013: Duplicate email handling:

- Check if email exists before creating accountĐ
- If exists, display error: "This email is already registered. Please sign in instead."Đ
- Provide "Forgot Password" link if user forgot passwordĐ
- Do not reveal if email exists to prevent email enumeration attacksĐ

FR-014: Password security:

- Hash password using bcrypt before storageĐ
- Never store plain text passwordĐ
- Never return password in API responsesĐ
- Implement password strength meter (visual indicator)Đ

3.3 Email Verification

FR-015: Verification email:

- Send email immediately after registrationÐ
- Email subject: "Verify your email address"Ð
- Email contains verification link with tokenÐ
- Link format: `https://app.example.com/verify-email?token={token}`Ð
- Link expires after 24 hoursÐ
- Email includes "Resend verification email" optionÐ

FR-016: Email verification process:

- User clicks verification link in emailÐ
- System validates token (exists, not expired, not used)Ð
- If valid, update user status to "verified"Ð
- Mark verification token as usedÐ
- Redirect to login page with success message: "Email verified successfully. Please sign in."Ð
- If invalid/expired, display error: "Verification link is invalid or expired. Please request a new verification email."Ð

FR-017: Resend verification email:

- User can request new verification emailÐ
- Generate new verification tokenÐ
- Invalidate old verification tokenÐ
- Send new verification emailÐ
- Limit resend to 3 times per hour per emailÐ
- Display message: "Verification email sent. Please check your inbox."Ð

FR-018: Unverified account restrictions:

- Unverified accounts cannot log inÐ
- Display message: "Please verify your email address before signing in. Check your inbox for verification link."Ð
- Allow resend verification email from login pageÐ
- Unverified accounts are deleted after 7 days if not verifiedÐ

3.4 Security Requirements

FR-019: Input sanitization:

- Sanitize all user inputs to prevent XSS attacksÐ
- Validate and escape special charactersÐ
- Prevent SQL injection through parameterized queriesÐ
- Block common attack patternsÐ

FR-020: Rate limiting:

- Limit registration attempts to 5 per hour per IP addressÐ
- Limit registration attempts to 3 per hour per email addressÐ
- Display error: "Too many registration attempts. Please try again later."Ð
- Implement CAPTCHA after 3 failed attemptsÐ

FR-021: Data protection:

- Encrypt sensitive data in transit (HTTPS)Ð

- Encrypt sensitive data at restÐ
- Comply with GDPR and CCPA regulationsÐ
- Allow users to delete their accounts and dataÐ

FR-022: Password requirements enforcement:

- Enforce password strength requirementsÐ
- Display password strength meterÐ
- Provide password requirements checklistÐ
- Prevent common weak passwords (e.g., "password123", "12345678")Ð

4. Non-Functional Requirements

4.1 Performance

NFR-001: Registration performance:

- Page load time: < 2 secondsÐ
- Form validation: < 100ms (real-time)Ð
- Account creation API: < 500ms (95th percentile)Ð
- Email sending: < 5 secondsÐ

NFR-002: Scalability:

- Support 1000+ concurrent registrationsÐ
- Handle traffic spikes during promotionsÐ

4.2 Usability

NFR-003: User experience:

- Clear, step-by-step registration flowÐ
- Real-time form validation with helpful error messagesÐ
- Progress indicator during submissionÐ
- Success confirmation with next stepsÐ
- Mobile-optimized registration formÐ

NFR-004: Accessibility:

- WCAG 2.1 AA complianceÐ
- Keyboard navigation supportÐ
- Screen reader compatibilityÐ
- Clear focus indicatorsÐ
- Color contrast ratio: minimum 4.5:1Ð

4.3 Security

NFR-005: Security standards:

- OWASP Top 10 security practicesÐ
- Regular security auditsÐ
- Vulnerability scanningÐ
- Penetration testingÐ

5. User Flows

5.1 Successful Registration Flow

1. User navigates to registration page
2. User enters first name: "John"
3. User enters last name: "Doe"
4. User enters email: "john.doe@example.com"
5. User enters password: "SecurePass123!"
6. User confirms password: "SecurePass123!"
7. User enters phone number (optional): "+1-555-123-4567"
8. User checks "I agree to Terms and Conditions"
9. User clicks "Create Account"
10. System validates all fields (all valid)
11. System creates user account
12. System sends verification email
13. System displays success message: "Account created! Please check your email to verify your account."
14. User receives verification email
15. User clicks verification link
16. System verifies email
17. System redirects to login page with success message

5.2 Registration with Validation Errors

1. User navigates to registration page
2. User enters invalid email: "invalid-email"
3. User enters weak password: "123"
4. User leaves first name empty
5. User clicks "Create Account"
6. System validates fields (multiple errors)
7. System displays errors:

- - "Please enter a valid email address"Đ
- - "Password must be at least 8 characters with uppercase, lowercase, number, and special character"Đ
- - "First name is required"Đ

8. User corrects all errors
9. User resubmits form
10. System validates (all valid)
11. System creates account

5.3 Duplicate Email Registration

1. User navigates to registration page
2. User enters email that already exists: "existing@example.com"
3. User fills all other fields correctly
4. User clicks "Create Account"
5. System checks for duplicate email

6. System displays error: "This email is already registered. Please sign in instead."
7. System provides "Sign in" link
8. User clicks "Sign in" link
9. User is redirected to login page

6. Technical Specifications

6.1 API Endpoints

POST /api/auth/register

• Request body: `{ "firstName": "John", "lastName": "Doe", "email": "john.doe@example.com", "password": "SecurePass123!", "confirmPassword": "SecurePass123!", "phone": "+15551234567", "dateOfBirth": "1990-01-01", "acceptTerms": true, "acceptMarketing": false }`Đ

• Success response (201): `{ "success": true, "message": "Account created. Please verify your email.", "userId": "uuid" }`Đ

• Error response (400): `{ "error": "Validation failed", "errors": { "email": "This email is already registered" } }`Đ

GET /api/auth/verify-email?token={token}

• Success response (200): `{ "success": true, "message": "Email verified successfully" }`Đ

• Error response (400): `{ "error": "Invalid or expired verification token" }`Đ

POST /api/auth/resend-verification

• Request body: `{ "email": "john.doe@example.com" }`Đ

• Success response (200): `{ "success": true, "message": "Verification email sent" }`Đ

6.2 Database Schema

users table:

• id (UUID, primary key)Đ
• first_name (VARCHAR(50))Đ
• last_name (VARCHAR(50))Đ
• email (VARCHAR(254), unique, indexed)Đ
• password_hash (VARCHAR(255))Đ
• phone (VARCHAR(20), nullable)Đ
• date_of_birth (DATE, nullable)Đ
• email_verified (BOOLEAN, default false)Đ
• verification_token (VARCHAR(255), nullable, indexed)Đ
• verification_token_expires_at (TIMESTAMP, nullable)Đ
• marketing_emails_opt_in (BOOLEAN, default false)Đ
• status (ENUM: 'pending_verification', 'verified', 'suspended', 'deleted')Đ
• created_at (TIMESTAMP)Đ
• updated_at (TIMESTAMP)Đ

7. Acceptance Criteria

1. ' User can successfully register with valid information
2. ' All validation errors display correctly
3. ' Duplicate email is detected and error displayed

4. ' Password strength requirements are enforced
5. ' Verification email is sent after registration
6. ' Email verification works correctly
7. ' Unverified accounts cannot log in
8. ' Resend verification email works
9. ' Rate limiting prevents abuse
10. ' All security requirements are implemented
11. ' Registration form is mobile-responsive
12. ' Form is accessible via keyboard and screen readers

8. Dependencies

- Email service (verification emails)Đ
- Database (user storage)Đ
- Password hashing library (bcrypt)Đ
- HTTPS certificate (SSL/TLS)Đ
- CAPTCHA service (abuse prevention)Đ
- Redis (rate limiting)Đ

9. Out of Scope

- Social registration (Google, Facebook) - future enhancementĐ
- Two-factor authentication - future enhancementĐ
- Profile picture upload - separate moduleĐ
- Address management - separate moduleĐ
- Account deletion - separate moduleĐ

10. Testing Requirements

- Unit tests for all validation logicĐ
- Integration tests for registration flowĐ
- End-to-end tests for email verificationĐ
- Security testing (XSS, SQL injection, rate limiting)Đ
- Performance testing (load testing)Đ
- Usability testing (user acceptance testing)Đ