Service and Utility Classes - UML Class Diagram
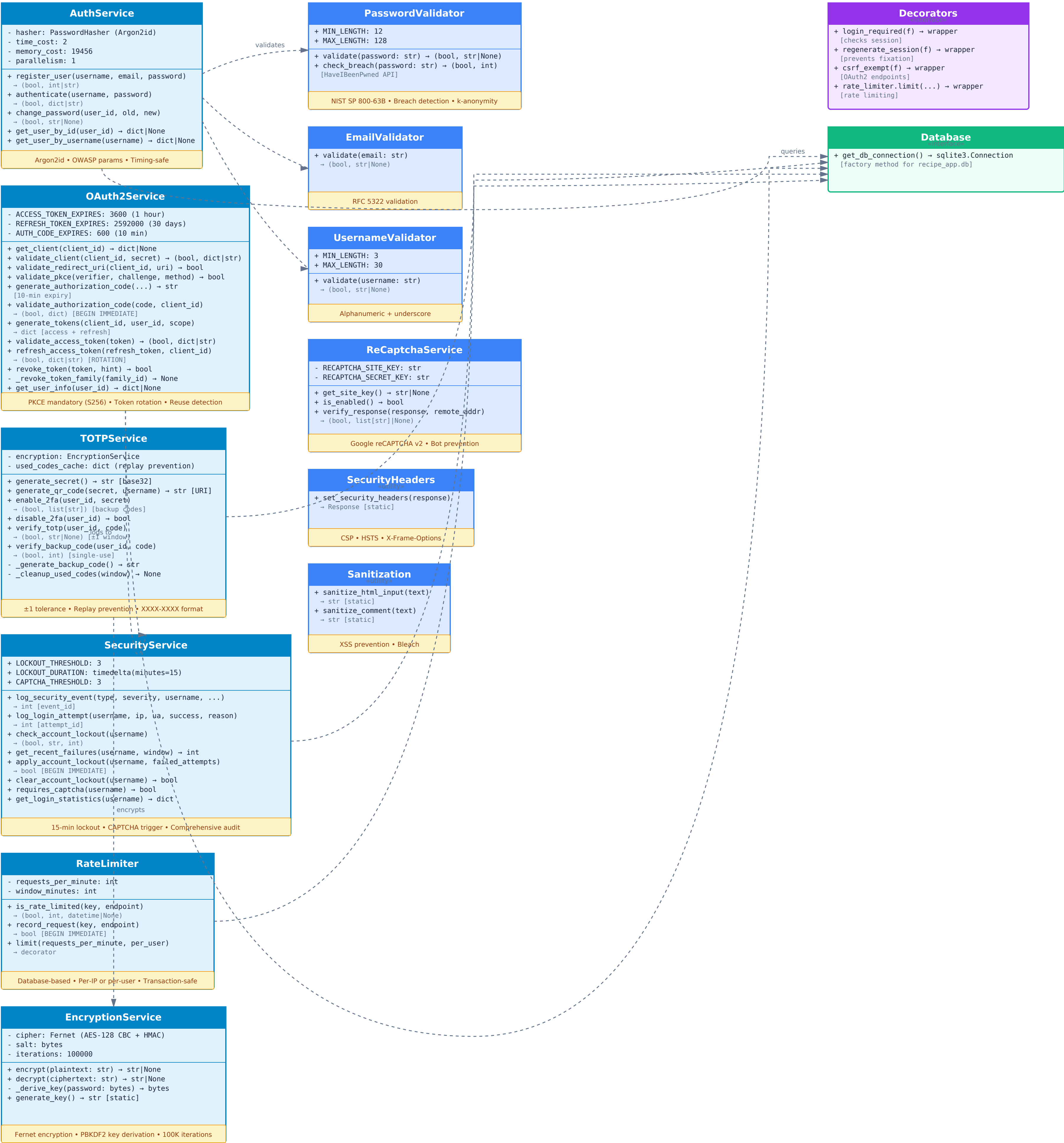
Organized layout: Services (left), Utilities (center), Database/Decorators (right)

AuthService

- hasher: PasswordHasher (Argon2id)
- time_cost: 2
- memory_cost: 19456
- parallelism: 1

+ register_user(username, email, password)
  → (bool, int|str)
+ authenticate(username, password)
  → (bool, dict|str)
+ change_password(user_id, old, new)
  → (bool, str|None)
+ get_user_by_id(user_id) → dict|None
+ get_user_by_username(username) → dict|None

Argon2id • OWASP params • Timing-safe

PasswordValidator

+ MIN_LENGTH: 12
+ MAX_LENGTH: 128

+ validate(password: str) → (bool, str|None)
+ check_breach(password: str) → (bool, int)
  [HaveIBeenPwned API]

NIST SP 800-63B • Breach detection • k-anonymity

Decorators

+ login_required(f) → wrapper
  [checks session]
+ regenerate_session(f) → wrapper
  [prevents fixation]
+ csrf_exempt(f) → wrapper
  [OAuth2 endpoints]
+ rate_limiter.limit(...) → wrapper
  [rate limiting]

EmailValidator

+ validate(email: str)
  → (bool, str|None)

RFC 5322 validation

Database

+ get_db_connection() → sqlite3.Connection
  [factory method for recipe_app.db]

OAuth2Service

- ACCESS_TOKEN_EXPIRES: 3600 (1 hour)
- REFRESH_TOKEN_EXPIRES: 2592000 (30 days)
- AUTH_CODE_EXPIRES: 600 (10 min)

+ get_client(client_id) → dict|None
+ validate_client(client_id, secret) → (bool, dict|str)
+ validate_redirect_uri(client_id, uri) → bool
+ validate_pkce(verifier, challenge, method) → bool
+ generate_authorization_code(...) → str
  [10-min expiry]
+ validate_authorization_code(code, client_id)
  → (bool, dict) [BEGIN IMMEDIATE]
+ generate_tokens(client_id, user_id, scope)
  → dict [access + refresh]
+ validate_access_token(token) → (bool, dict|str)
+ refresh_access_token(refresh_token, client_id)
  → (bool, dict|str) [ROTATION]
+ revoke_token(token, hint) → bool
+ _revoke_token_family(family_id) → None
+ get_user_info(user_id) → dict|None

PKCE mandatory (S256) • Token rotation • Reuse detection

UsernameValidator

+ MIN_LENGTH: 3
+ MAX_LENGTH: 30

+ validate(username: str)
  → (bool, str|None)

Alphanumeric + underscore

ReCaptchaService

- RECAPTCHA_SITE_KEY: str
- RECAPTCHA_SECRET_KEY: str

+ get_site_key() → str|None
+ is_enabled() → bool
+ verify_response(response, remote_addr)
  → (bool, list[str]|None)

Google reCAPTCHA v2 • Bot prevention

TOTPService

- encryption: EncryptionService
- used_codes_cache: dict (replay prevention)

+ generate_secret() → str [base32]
+ generate_qr_code(secret, username) → str [URI]
+ enable_2fa(user_id, secret)
  → (bool, list[str]) [backup codes]
+ disable_2fa(user_id) → bool
+ verify_totp(user_id, code)
  → (bool, str|None) [±1 window]
+ verify_backup_code(user_id, code)
  → (bool, int) [single-use]
- _generate_backup_code() → str
- _cleanup_used_codes(window) → None

±1 tolerance • Replay prevention • XXXX-XXXX format

SecurityHeaders

+ _set_security_headers(response)
  → Response [static]

CSP • HSTS • X-Frame-Options

Sanitization

+ sanitize_html_input(text)
  → str [static]
+ sanitize_comment(text)
  → str [static]

XSS prevention • Bleach

SecurityService

+ LOCKOUT_THRESHOLD: 3
+ LOCKOUT_DURATION: timedelta(minutes=15)
+ CAPTCHA_THRESHOLD: 3

+ log_security_event(type, severity, username, ...)
  → int [event_id]
+ log_login_attempt(username, ip, ua, success, reason)
  → int [attempt_id]
+ check_account_lockout(username)
  → (bool, str, int)
+ get_recent_failures(username, window) → int
+ apply_account_lockout(username, failed_attempts)
  → bool [BEGIN IMMEDIATE]
+ clear_account_lockout(username) → bool
+ requires_captcha(username) → bool
+ get_login_statistics(username) → dict

15-min lockout • CAPTCHA trigger • Comprehensive audit

RateLimiter

- requests_per_minute: int
- window_minutes: int

+ is_rate_limited(key, endpoint)
  → (bool, int, datetime|None)
+ record_request(key, endpoint)
  → bool [BEGIN IMMEDIATE]
+ limit(requests_per_minute, per_user)
  → decorator

Database-based • Per-IP or per-user • Transaction-safe

EncryptionService

- cipher: Fernet (AES-128 CBC + HMAC)
- salt: bytes
- iterations: 100000

+ encrypt(plaintext: str) → str|None
+ decrypt(ciphertext: str) → str|None
- _derive_key(password: bytes) → bytes
+ generate_key() → str [static]

Fernet encryption • PBKDF2 key derivation • 100K iterations

Flask Blueprints (Routes Layer)

• AuthBlueprint (/auth/*) → AuthService, SecurityService, RateLimiter, ReCaptchaService

• OAuthBlueprint (/oauth/*) → OAuth2Service, SecurityService

• TwoFABlueprint (/2fa/*) → TOTPService, AuthService, SecurityService, RateLimiter

UML Notation Legend

Visibility:
+ Public - Private

Relationships:
- - - → Dependency (uses)

Stereotypes:
«interface» - Protocol | «utility» - Static class

Security Annotations:
[BEGIN IMMEDIATE] Transaction-protected
[ROTATION] Token family rotation

Class Types:
☐ Service ☐ Utility ☐ Decorator

v2.0 - Generated from ground truth analysis - Clean layout with organized columns and labeled arrows