User Management System: Software Requirement Specification (SRS) - Version 2.0

Document Version: 2.0.0

Date: December 12, 2025

System: User Management Service (V2 - Authenticated Profiles)

Backend Framework: Python (Flask/Similar)

Database: SQLite

I. General System Requirements

Requirement ID	Description
G-001	The backend must be implemented using a Python-based framework.
G-002	The database used for persistence must be SQLite.
G-003	The solution must be containerized using Docker.

G-004	**No password encryption is required for this implementation.**
G-005	**All secure endpoints (profile CRUD, password reset, logout) must require a valid session_key.**
G-006	**Email field is the unique identifier and cannot be updated via the API.**

II. Database Schema (SQLite)

1. user_mngt_users Table (User Profiles)

Field Name	**Data Type**	**Constraint**	**Description**
id	INTEGER	PRIMARY KEY AUTOINCREMENT	Unique user identifier.
email	TEXT	NOT NULL, UNIQUE, 5-25 chars	User's email (login ID).
password	TEXT	NOT NULL, Exactly 6 chars	User's password.
confirmation_token	TEXT	UNIQUE, Nullable	Token for email confirmation.

is_confirmed	INTEGER	NOT NULL, DEFAULT 0	Confirmation status (0: Pending, 1: Confirmed).
first_name	TEXT	**NOT NULL**, 5-25 chars, **Alphabetic only**	User's first name.
last_name	TEXT	**NOT NULL**, 5-25 chars, **Alphabetic only**	User's last name.
address1	TEXT	Nullable, 5-25 chars	Primary address line.
address2	TEXT	Nullable, 5-25 chars	Secondary address line.
city	TEXT	Nullable, 5-25 chars	City.
state	TEXT	Nullable, 5-25 chars	State/Region.
country	TEXT	**NOT NULL**, 5-25 chars	Country.
pin_code	TEXT	**NOT NULL**, 5-25 chars	Postal/Pin Code.
contact_country_code	TEXT	**NOT NULL**, Exactly 3 digits	Country calling code (e.g., '91', '001').
contact_number	TEXT	**NOT NULL**, Exactly 10 digits	Main contact number.

created_at	TEXT	NOT NULL	Timestamp of record creation.

2. user_mngt_sessions Table (Session Management)

Field Name	**Data Type**	**Constraint**	**Description**
session_key	TEXT	PRIMARY KEY	Unique UUID generated upon login.
user_id	INTEGER	FOREIGN KEY	Links to user_mngt_users.id.
is_active	INTEGER	NOT NULL, DEFAULT 1	Session status (1: Active, 0: Logged out).
created_at	TEXT	NOT NULL	Session start time.

III. API Endpoint Specifications

1. API: System Version

Field	**Value**
ID	API-001
Endpoint	/version
Method	GET
Success Body	{"version": "2.0.0", "service": "User-Management-Service"}

2. API: User Registration (Pre-Confirmation)

Field	Value
ID	API-002
Endpoint	/register
Method	POST
Input (Body)	email, password (Mandatory, V1 validation)
Success Body	{"message": "Registration successful, please click the confirmation link.", "confirmation_link": "..."}

3. API: Registration Confirmation

Field	Value
ID	API-003
Endpoint	/confirm_registration/{token}
Method	GET
Input (Path)	token (Mandatory)
Success Body	{"message": "Registration successfully confirmed. You can now login."}

4. API: User Login (UPDATED)

Field	**Value**
ID	API-004
Endpoint	/login
Method	POST
Input (Body)	email, password (Mandatory)
Process	Authenticate credentials. **Generate and store unique session_key**.
Success Code	200
Success Body	{"message": "Login successful.", "user_id": 123, "session_key": "generated_uuid_session_key"}
Failure Code	401/403

5. API: User Logout (NEW)

Field	**Value**
ID	API-005

Endpoint	/logout
Method	POST
Input (Body)	email, session_key (Mandatory)
Process	Validate credentials. **Deactivate session** (is_active=0).
Success Code	200
Success Body	{"message": "Logout successful."}
Failure Code	401
Failure Body	{"message": "Invalid email or session key."}

6. API: Get User Details (NEW)

Field	**Value**
ID	API-006
Endpoint	/user_mngt_user
Method	GET

Input (Query)	email, session_key (Mandatory)
Process	Authenticate session. Retrieve all user profile fields.
Success Code	200
Success Body	JSON object containing all 12 user fields (excluding password).
Failure Code	401
Failure Body	{"message": "Invalid session or user not found."}

7. API: Update User Details (NEW)

Field	**Value**
ID	API-007
Endpoint	/user_mngt_user
Method	PUT
Input (Query)	email, session_key (Mandatory)
Input (Body)	All 12 profile fields (Mandatory fields must be present).

Validation	**First/Last Name:** 5-25 chars, Alphabetic. **Contact Code:** Exactly 3 digits. **Contact Number:** Exactly 10 digits.
Success Code	200
Success Body	{"message": "User details successfully updated."}
Failure Code	400/401
Failure Body	{"message": "Invalid input format or session."}

8. API: Delete User (NEW)

Field	**Value**
ID	API-008
Endpoint	/user_mngt_user
Method	DELETE

Input (Body)	email, password, session_key (Mandatory)
Process	Authenticate session and password. **Delete user and all sessions.**
Success Code	200
Success Body	{"message": "User account successfully deleted."}
Failure Code	401
Failure Body	{"message": "Invalid email, password, or session key."}

9. API: Forget Password / Update Password (NEW)

Field	**Value**
ID	API-009
Endpoint	/forget_password
Method	PUT
Input (Body)	email, new_password, confirm_new_password, session_key (Mandatory)
Validation	Passwords must match and be exactly 6 characters.

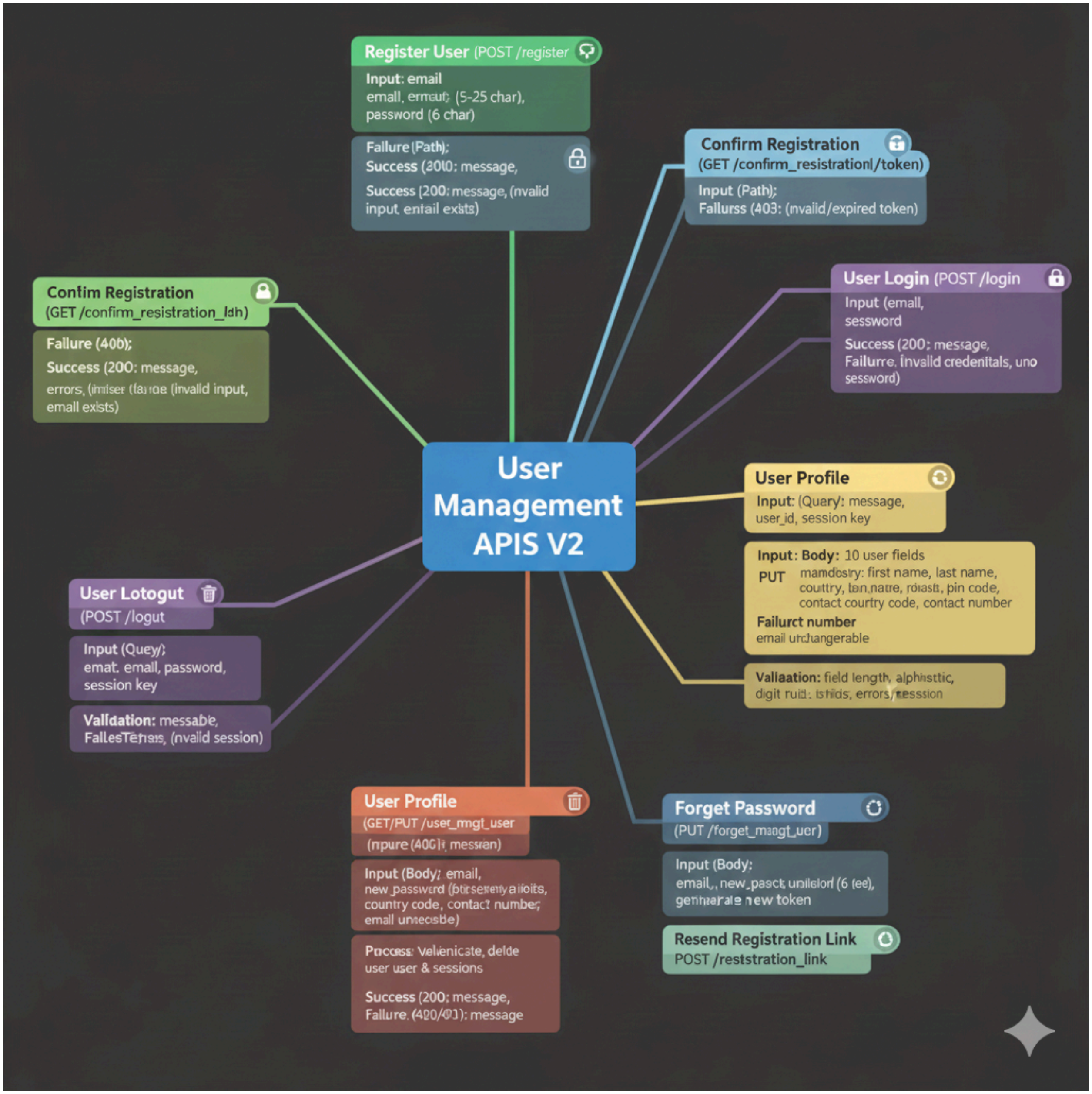
Process	Authenticate session. Update user's password.
Success Code	200
Success Body	{"message": "Password successfully updated."}
Failure Code	400/401
Failure Body	{"message": "Passwords do not match or Invalid session/email."}

10. API: Resend Registration Link (NEW)

Field	**Value**
ID	API-010
Endpoint	/resend_registration_link
Method	POST
Input (Body)	email, password (Mandatory)
Process	Validate credentials. If user is not confirmed, generate and store a new token, then return the link.
Success Code	200

Success Body	{"message": "New registration link generated.", "confirmation_link": "..."}
Failure Code	400/401
Failure Body	{"message": "Invalid email/password or account is already confirmed."}

III. Mind Map


Register User (POST /register
Input: email
emall, ermaut; (5-25 char),
password (6 char)
Fallure (Path);
Success (2010: message,
Success (200: message, (nvalid
input, entail exsts)
Confirm Registration
(GET /confirm_resistrationl/token)
Input (Path);
Fallurss (403: (nvalid/expired token)
Confim Registration
(GET /confirm_resistration_ldh)
Fallure (40b);
Success (200: message,
errors, (imiser (la:ıoe (invalid input,
emall exists)
User Login (POST /login
Input (emall,
sessword
Success (200: message,
Failurre. Invalid credenitals, uno
sessword)
User
Management
APIS V2
User Profile
Input: (Query: message,
user_id, session key
Input: Body: 10 user fields
PUT
mamdbstry: first name, last name,
couttry, tan.natre, roiastt, pin code,
contact courtry code, contact number
Failurct number
email urchangerable
Valiaation: field length, alphnsttic,
digit ruid: ishids, errors, session
User Lotogut
(POST /logut
Input (Quey);
emat. emall, password,
session key
Validation: messable,
FallesTeneis, (nvalid session)
User Profile
(GET/PUT /user_mgt_user
(npure (400]i, messran)
Input (Body; email,
new_passwcrd (btrsenetyaiioits,
country code, contact number;
email umecsbe)
Process: Velienicate, delde
user user & sessions
Success (200: message,
Fallure. (420/401): message
Forget Password
(PUT /forget_maagt_uer)
Input (Body:
email,, new_pasct unilslorl (6 (ee),
genrarale new token
Resend Registration Link
POST /reststration_link