Software Requirements Specification (SRS)

MemoLink

1. Introduction

1.1 Purpose

The purpose of this document is to specify the functional and non-functional requirements for the "MemoLink" software. This SRS describes *what* the system will do, not *how* it will be built. It serves as a contract between the development team and the "customer" (instructor) to ensure a shared understanding of the system's scope and deliverables.

1.2 Document Conventions

- **FR:** Functional Requirement
- **NFR:** Non-Functional Requirement
- **US:** User Story
- **MoSCoW:** Prioritization method (Must have, Should have, Could have, Won't have)

1.3 Intended Audience

This document is intended for project developers, project managers, and the project instructor (customer).

1.4 Project Scope

This system will allow users to create, edit, delete, and view "Memory Packages" (Nodes) containing text and images. Users will be able to create bidirectional links between these nodes. The system will also provide an interactive graph visualization of the user's entire memory network.

Out of Scope:

- Real-time collaboration between users.
- Social media integration or sharing.
- A dedicated native mobile application.

1.5 References

- D1 - Software Development Plan (SDP):

2. Overall Description

2.1 Product Perspective

The system is a stand-alone web application. It relies on a backend API (e.g., Python/FastAPI) to serve the API and a database (e.g., PostgreSQL) for data persistence. The frontend (e.g., React) communicates with the backend via a REST API. The system will be deployed on a cloud platform (e.g., Render).

2.2 Product Functions

A summary of the major functions the system will perform:

- User registration and authentication.
- Create, Read, Update, and Delete (CRUD) operations for Memory Nodes.
- Creation and deletion of links between Memory Nodes.
- Interactive graph visualization of the node network.
- Grid-based "Album" view of all user memories.

2.3 User Classes and Characteristics

There is one primary user class:

- **Registered User:** Any individual who has created an account. These users are expected to have basic web literacy. Their primary need is to digitally organize, store, and find relationships between their personal memories, notes, or ideas.

2.4 Operating Environment

- **Backend:** Will run in a Linux environment of a cloud platform.
- **Frontend:** Will operate in modern web browsers (latest two major versions of Chrome, Firefox, and Safari).

2.5 Design and Implementation Constraints

- The backend API **must** be developed using Python (FastAPI).

- The frontend **must** be developed using React.js.
- The database **must** be PostgreSQL.
- The system **must** be deployable on a cloud platform.
- All communication between frontend and backend **must** be over HTTPS.

2.C Assumptions and Dependencies

- Users are assumed to have a stable internet connection.
- The system is dependent on the uptime and services provided by the hosting platform.
- The system is not dependent on any external third-party APIs for its core functionality (e.g., image storage) at this stage.

3. Functional Requirements

3.1 User Stories

US-01: New User Registration

- **As a** new visitor
- **I want to** register for an account using my name, surname, email, and a password
- **So that** I can start creating and saving my own memory packages.
- **Priority:** Must Have
- **Acceptance Criteria:**
  - **Given** I am on the Registration page
  - **When** I fill in my name, surname, a valid email, and a password (min. 8 chars) and click "Register"
  - **Then** my account is created AND I am redirected to the Login page.
  - **Given** I am on the Registration page
  - **When** I enter an email that already exists in the system
  - **Then** I see an error message "This email is already in use".

US-02: User Login

- **As a** registered user
- **I want to** log in with my email and password
- **So that** I can access my memories.
- **Priority:** Must Have
- **Acceptance Criteria:**
  - **Given** I am on the Login page
  - **When** I enter my correct email and password and click "Login"
  - **Then** I am authenticated AND redirected to my Dashboard.
  - **Given** I am on the Login page
  - **When** I enter an incorrect password
  - **Then** I see an error message "Invalid email or password".

US-03: Create New Memory Node

- **As a** registered user
- **I want to** create a new "Memory Package" (Node) with a title and a description
- **So that** I can save a thought, memory, or piece of information.
- **Priority:** Must Have
- **Acceptance Criteria:**
  - **Given** I am logged in
  - **When** I open the "Create New Memory" form, enter a title and description, and click "Save"
  - **Then** the new memory node is created AND I can see it in my 'Album' view.
  - **Given** I am on the "Create New Memory" form
  - **When** I try to save with an empty title field
  - **Then** I see an error "Title is a required field" AND the node is not created.

US-04: View All Memories in Album

- **As a** registered user
- **I want to** see all my memories in a grid format on the 'Album' page
- **So that** I can quickly browse all my saved content.
- **Priority:** Must Have
- **Acceptance Criteria:**
  - **Given** I am logged in
  - **When** I click the 'Album' link in the navigation bar
  - **Then** I see all my memory nodes displayed as cards, showing their title and (if present) image.

US-05: Create Narrative Connection

- **As a** registered user
- **I want to** create a "Narrative Connection" between two memory nodes
- **So that** I can visually connect related memories or ideas.
- **Priority:** Must Have
- **Acceptance Criteria:**
  - **Given** I am on the 'Graph' view and have at least two nodes
  - **When** I click on the first node, select the "Create Connection" action, and then click the second node
  - **Then** a visual line (connection) is drawn between the two nodes AND this relationship is saved to the database.

US-06: View Memory Graph

- **As a** registered user
- **I want to** view all my nodes and their links as an interactive network graph on the 'Your Graph' page
- **So that** I can explore the high-level relationships between my memories.
- **Priority:** Must Have
- **Acceptance Criteria:**
  - **Given** I am logged in and open the 'Your Graph' page
  - **When** The page loads
  - **Then** All my nodes are rendered as points AND all links are rendered as lines connecting them.
  - **Given** I am on the 'Graph' view
  - **When** I click and drag a node with my mouse
  - **Then** I can move the node and its connected links around the screen.

US-07: Update Memory Node

- **As a** registered user
- **I want to** edit the title and description of a memory node I created
- **So that** I can correct mistakes or add more information.
- **Priority:** Must Have
- **Acceptance Criteria:**
  - **Given** I am viewing the details of my node (e.g., in a modal or on its page)
  - **When** I click the "Edit" button
  - **Then** the title and description fields become editable.
  - **Given** I am editing a node
  - **When** I change the description text and click "Save"

    - **Then** the new description is saved to the database AND I see a "Save Successful" message.

US-08: Delete Memory Node

- **As a** registered user
- **I want to** permanently delete a memory node
- **So that** I can remove irrelevant or unwanted information.
- **Priority:** Must Have
- **Acceptance Criteria:**
    - **Given** I am viewing a node I own
    - **When** I click the "Delete" button
    - **Then** I am shown a confirmation prompt (e.g., "Are you sure you want to delete this node?").
    - **Given** I have clicked "Delete"
    - **When** I confirm "Yes" at the prompt
    - **Then** the node is permanently removed from the system AND it disappears from my 'Graph' and 'Album' views.
    - **Given** a node is deleted
    - **When** I view the graph
    - **Then** all links that were attached to that node are also deleted.

US-0G: Add Image to Node

- **As a** registered user
- **I want to** upload an image or add an image URL when creating or editing a node
- **So that** I can add visual context to my memories.
- **Priority:** Should Have
- **Acceptance Criteria:**
    - **Given** I am on the "Create New Memory" form
    - **When** I upload a valid image file (JPEG/PNG) and click "Save"
    - **Then** the node is created with the image attached.
    - **Given** I am on the "Create New Memory" form
    - **When** I paste a valid external image URL into the 'Image URL' field and click "Save"
    - **Then** the node is created and displays the external image.
    - **Given** I try to upload a file larger than the (e.g., 5MB) size limit
    - **Then** I see an error message "File size exceeds limit".

US-10: View Node Details

- **As a** registered user

- **I want to** click on a node in the Album or Graph view
- **So that** I can see its full description and a list of its direct connections.
- **Priority:** Must Have
- **Acceptance Criteria:**
    - **Given** I am on the 'Album' page
    - **When** I click on a memory card
    - **Then** a detail view (modal or new page) opens showing the full title and description.
    - **Given** I am in the node detail view
    - **When** the node is linked to other nodes
    - **Then** I see a list of "Incoming Links" and "Outgoing Links" by their titles.

US-11: Sort and Paginate Album View

- **As a** registered user
- **I want to** sort my memories in the 'Album' view by date or title
- **So that** I can find specific memories more easily.
- **Priority:** Should Have
- **Acceptance Criteria:**
    - **Given** I am on the 'Album' page
    - **When** I select "Sort by Title (A-Z)" from a dropdown
    - **Then** the grid of memories re-orders alphabetically by title.
    - **Given** I have more than (e.g., 30) memories
    - **When** I am on the 'Album' page
    - **Then** I see pagination controls (e.g., "Page 1 of 3") to navigate between pages.

US-12: Delete Node Link

- **As a** registered user
- **I want to** remove a link between two memory nodes
- **So that** I can correct relationships that are no longer accurate.
- **Priority:** Must Have
- **Acceptance Criteria:**
    - **Given** I am on the 'Graph' view
    - **When** I right-click on a link (the line) between two nodes
    - **Then** I see a "Delete Link" option.
    - **Given** I have clicked "Delete Link"
    - **When** I confirm the action
    - **Then** the link is removed from the database AND the line disappears from the graph.

US-13: View Dashboard

- **As a** registered user
- **I want to** see a dashboard when I first log in
- **So that** I can get a quick overview of my memory network.
- **Priority:** Should Have
- **Acceptance Criteria:**
    - **Given** I have just logged in
    - **When** I am on the Dashboard page
    - **Then** I see a statistic for "Total Memories Created".
    - **Given** I have just logged in
    - **When** I am on the Dashboard page
    - **Then** I see a statistic for "Total Links Created".
    - **Given** I have created memories
    - **When** I am on the Dashboard
    - **Then** I see a list of my "5 Most Recently Added Memories" with links to view them.

US-14: View User Profile

- **As a** registered user
- **I want to** view my profile page
- **So that** I can see my account details (name, surname, email).
- **Priority:** Could Have
- **Acceptance Criteria:**
    - **Given** I am logged in
    - **When** I click 'Profile' in the user menu
    - **Then** I am taken to my profile page AND I can see my registered name, surname, and email address.
    - **Given** I am logged in as User A
    - **When** I try to navigate directly to the profile URL of User B
    - **Then** I see a "Forbidden" error (403) OR I am redirected to my own dashboard.

US-15: Update User Profile

- **As a** registered user
- **I want to** update my profile information (name, surname)
- **So that** I can keep my account details accurate.
- **Priority:** Could Have
- **Acceptance Criteria:**

- **Given I** am on my profile page
    - **When I** click "Edit Profile", change my first name, and click "Save"
    - **Then** my name is updated in the database AND I see a "Profile Updated" success message.
    - **Given I** am on my profile page
    - **When I** try to change my email address
    - **Then** the email field is read-only (not editable).
    - **Given I** am editing my profile
    - **When I** try to save my name as a blank field
    - **Then** I see an error message "Name cannot be empty".

US-16: Reset Password

- **As a** registered user
- **I want to** reset my password if I forget it
- **So that** I can regain access to my account.
- **Priority:** Must Have
- **Acceptance Criteria:**
    - **Given I** am on the Login page
    - **When I** click the "Forgot Password?" link
    - **Then I** am taken to a page asking for my email.
    - **Given I** have entered my registered email
    - **When I** submit the form
    - **Then I** receive an email with a unique password reset link.
    - **Given I** have clicked the reset link in my email
    - **When I** enter a new valid password and confirm it
    - **Then** my password is updated AND I am redirected to the Login page.

3.3 Functional Requirements List

3.3.1 Authentication s Accounts

- **FR-1:** The system shall allow users to register with name, surname, email, and password.
- **FR-2:** The system shall hash passwords and store only the hash.
- **FR-3:** The system shall allow users to log in and receive an access token.
- **FR-4:** The system shall require a valid token for protected endpoints.
- **FR-5:** The system shall provide a "Forgot Password" link on the Login page.
- **FR-6:** The system shall allow a user to reset their password via an email link.

3.3.2 User Profile

- **FR-7:** The system shall let users view their own profile.
- **FR-8:** The system shall let users update their profile fields, subject to validation.
- **FR-G:** The system shall prevent access to other users' profiles.

3.3.3 Nodes (Memories)

- **FR-10:** The system shall let authenticated users create nodes with title and description.
- **FR-11:** The system shall let users read, update, and delete their own nodes.
- **FR-12:** The system shall list a user's nodes with pagination and sorting.

3.3.4 Node Links

- **FR-13:** The system shall let users create and delete links between nodes they can access.
- **FR-14:** The system shall list incoming and outgoing links for a node.

3.3.5 Images

- **FR-15:** The system shall accept image uploads.
- **FR-16:** The system shall serve images by image_id and reject files over the configured size or with invalid MIME types.

3.3.6 API Behavior

- **FR-17:** The system shall return JSON responses with appropriate HTTP status codes.
- **FR-18:** The system shall expose a health endpoint reporting service status.

3.3.7 User Interface Views

- **FR-1G:** The system shall provide a Login view that allows the user to input their email address and password in designated fields.
- **FR-20:** The system shall provide a Register view that allows the user to input their name, surname, email address and password in designated fields for account creation.
- **FR-21:** The system shall provide a Dashboard view that displays recently added memories and key statistics (total memories, total connections).
- **FR-22:** The system shall provide a dynamic 'Your Graph' view that renders all linked memories as an interactive graph.
- **FR-23:** The system shall provide an 'Album' view that displays all memories in a grid format.

4. Non-Functional Requirements

4.1 Performance Requirements

- **NFR-1:** Average API response time expected should be below 1 second on the hosting platform under normal load.
- **NFR-2:** The system shall support at least 20 concurrent users.

4.2 Security Requirements

- **NFR-3:** User passwords shall be hashed before storage.
- **NFR-4:** Endpoints shall use token-based authentication.
- **NFR-5:** Communication with the deployed service shall occur over HTTPS.

4.3 Usability Requirements

- **NFR-6:** The API shall return JSON responses with clear field names.
- **NFR-7:** Swagger / OpenAPI documentation shall be automatically available.

- **NFR-8:** Clarity: The interface must clearly show where a memory is located: in the Dashboard, the Album, or the Graph, via a clear Navigation Bar.

4.4 Reliability and Availability

- **NFR-G:** The service shall achieve at least 95% uptime on the hosting platform.
- **NFR-10:** Database backups shall allow full restoration of data.

4.5 Maintainability Requirements

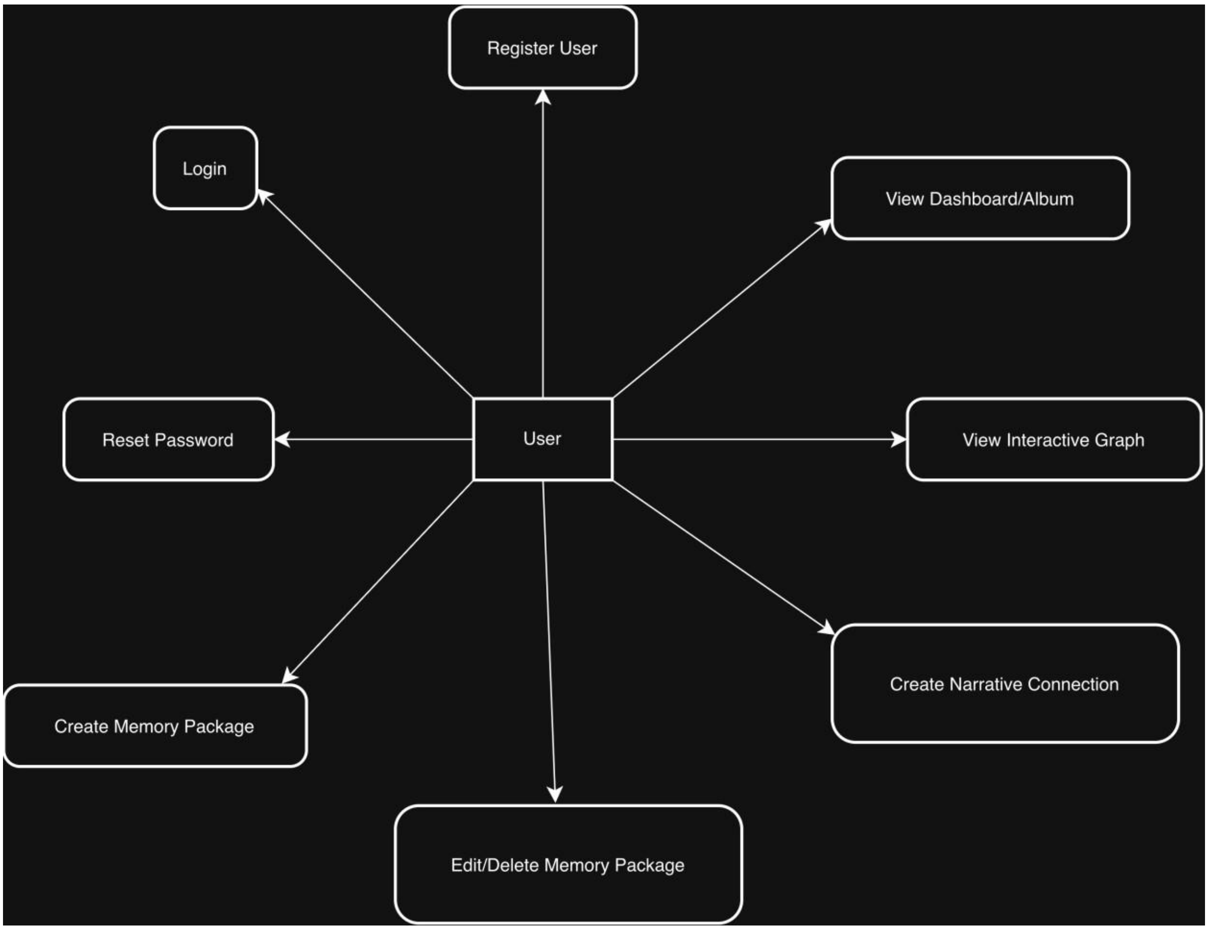
- **NFR-11:** Code shall follow PEP 8 style and be version-controlled with Git.
- **NFR-12:** Application layers (routes, models, database) shall be modular.

4.C Compatibility Requirements

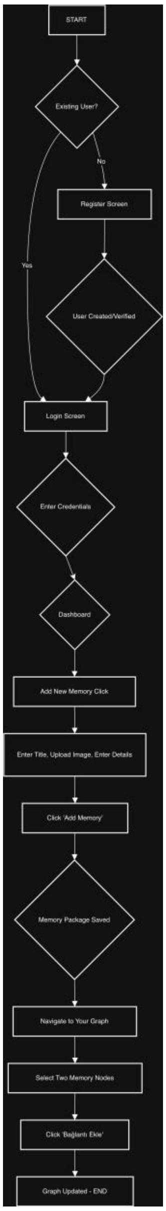
- **NFR-13:** The system shall run on hosting platform's Linux environment with Python 3.11+
  and PostgreSQL 15+.
- **NFR-14:** The API shall be testable from any modern browser or REST client.
- **NFR-15:** Browser Compatibility: The frontend (React.js) must be compatible with the current and immediately preceding major versions of Chrome, Safari, and Firefox.
- **NFR-16:** API Protocol: All communication shall use REST over HTTPS for standard compatibility.

5. System Models (Diagrams)

5.1 Use Case Diagram



5.2 User Flow Diagram


START
Existing User?
No
Yes
Register Screen
User Created/Verified
Login Screen
Enter Credentials
Dashboard
Add New Memory Click
Enter Title, Upload Image, Enter Details
Click 'Add Memory'
Memory Package Saved
Navigate to Your Graph
Select Two Memory Nodes
Click 'Bağlantı Ekle'
Graph Updated - END