

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN**



Software Requirements Specifications for Family Tree Application

Version 1.0 approved

Prepared by

Nguyễn Minh Hiếu – 23120124

Table of Contents

Table of Contents	ii
Revision History	iii
1. Introduction	1
1.1 Purpose	1
1.2 Document Conventions	1
1.3 Project Scope	1
1.4 References	1
2. Overall Description	2
2.1 Product Perspective	2
2.2 User Classes and Characteristics	2
2.3 Operating Environment	2
2.4 Design and Implementation Constraints	2
2.5 Assumptions and Dependencies	2
3. System Features	3
3.1 Member Management	3
3.1.1 Description	3
3.1.2 Stimulus/Response Sequences	3
3.1.3 Functional Requirements	3
3.2 Achievement Recording	3
3.2.1 Description	3
3.2.2 Functional Requirements	3
3.3 Member Search	4
3.3.1 Description	4
3.3.2 Functional Requirements	4
3.4 Report Generation	4
3.4.1 Description	4
3.4.2 Functional Requirements	4
3.5 Rule Modification	4
3.5.1 Description	4
3.5.2 Functional Requirements	4
4. Data Requirements	5
4.1 Logical Data Model	5
4.2 Data Dictionary	6
4.3 Reports	8
4.3.1 Report 1: Member Growth and Decline Summary (BM5.1)	8
4.3.2 Report 2: Member Achievements Summary (BM5.2)	9
4.4 Data Acquisition, Integrity, Retention, and Disposal	9
4.4.1 Data Acquisition	9
4.4.2 Data Integrity	9
4.4.3 Data Retention and Backup	10
4.4.4 Data Disposal	11
5. External Interface Requirements	11
5.1 User Interfaces	11
5.2 Communications Interfaces	11
6. Quality Attributes	11
6.1 Usability	11
6.2 Performance	11
6.3 Security	12
6.4 Safety	12
7. Other Requirements	12
Appendix A: Glossary	12

Appendix B: Analysis Models12

Revision History

Name	Date	Reason For Changes	Version
Nguyễn Minh Hiếu	12/10/2025	Initial version	1.0

1. Introduction

The introduction presents an overview to help the reader understand how the SRS is organized and how to use it.

1.1 Purpose

This document specifies the software requirements for the product "**Family Tree Management Software**", version 1.0. This document is intended for the following types of readers:

- **Project Managers:** To clearly understand the features to be built.
- **Developers:** To clearly understand the features to be built.
- **Testers:** To create test cases.
- **End-Users:** To know the scope and functionality of the software.

1.2 Document Conventions

This document uses standard conventions. Each functional requirement will be identified by a unique ID in the format **REQ-FN-XXX** for easy tracking and reference.

1.3 Project Scope

The "**Family Tree Management Software**" is developed to digitize the storage, management, and retrieval of information about members within a family line.

The main features include:

- **Receiving and managing member information:** Storing detailed information about new members, including family relationships (child, spouse).
- **Recording important events:** Allowing the recording of members' achievements and death events.
- **Information lookup:** Easily searching for and viewing detailed information of members in the family tree.
- **Report generation:** Generating statistical reports on the increase/decrease of members and achievements by year.
- **Customizing rules:** Allowing the administrator to change pre-defined categories such as hometowns, occupations, causes of death, etc.

1.4 References

"Project List" document ([Link](#)), section 1.9: "FAMILY TREE MANAGEMENT PROJECT".

2. Overall Description

This section presents a high-level overview of the product and the environment in which it will be used, the anticipated users, and known constraints, assumptions, and dependencies.

2.1 Product Perspective

This website offers a modern solution for managing family trees, replacing manual approaches like handwritten notes or scattered digital documents.

2.2 User Classes and Characteristics

- **Administrator:** The person primarily responsible for entering and maintaining the family tree data. This user has full rights to add, edit, and delete member information, generate reports, and change system rules.
- **Viewer:** Other family members who need to view and look up information. This user class only has viewing permissions and cannot edit data.

2.3 Operating Environment

- **Platform:** Web Application (Website).
- **User Browsers:** The website must be compatible with the latest versions of Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge.
- **Server Environment:**
 - **Operating System:** Linux.
 - **Web Server:** Azure.
 - **Database:** MySQL.
 - **Storage:** Data is stored centrally in the server's database.

2.4 Design and Implementation Constraints

- Users must have an internet connection to access and use the website.
- The website's interface language must be English.
- The website must have a responsive design to ensure proper display on various devices such as desktops, tablets, and mobile phones.

2.5 Assumptions and Dependencies

- It is assumed that the administrator has basic office computer skills.
- The input data (information about members) is assumed to be accurate.

3. System Features

This section details the functional requirements of the system.

3.1 Member Management

3.1.1 Description

This feature allows the Administrator to add new members, update information, and record the death of a member. Priority: **High**.

3.1.2 Stimulus/Response Sequences

- User selects the "Add Member" function.
- The system displays the "Member Profile" form (BM1).
- The user enters information and clicks "Save".
- The system validates the data and saves it to the database.

3.1.3 Functional Requirements

- **REQ-FN-001:** The system must provide a form to enter new member information according to the fields in **BM1: Member Profile**.
- **REQ-FN-002:** The system must comply with **QD1**, including allowing only the selection of relationship types (Child, Spouse) and validating dates (relationship date, birth date) according to natural laws.
- **REQ-FN-003:** The system must provide a pre-defined list of 4 hometowns and 15 occupations for the user to select from.
- **REQ-FN-004:** The system must provide a form to record a member's passing according to the fields in **BM4: Death Record**.
- **REQ-FN-005:** The system must comply with **QD4**, including checking that the time of death is logically after the date of birth and providing a pre-defined list of 12 causes of death and 3 burial locations.

3.2 Achievement Recording

3.2.1 Description

This feature allows for the recording of achievements obtained by members. Priority: **Medium**.

3.2.2 Functional Requirements

- **REQ-FN-006:** The system must provide a form to enter achievement information according to **BM2: Achievement Record**.
- **REQ-FN-007:** The system must comply with **QD2** by providing a list of 10 types of achievements for the user to choose from.

3.3 Member Search

3.3.1 Description

This feature allows users to search for and view a list of members. Priority: **High**.

3.3.2 Functional Requirements

- **REQ-FN-008:** The system must allow searching for members by criteria such as Full Name, Generation, and Father/Mother.
- **REQ-FN-009:** The search results must be displayed in a list format according to the structure of **BM3: Member List**.

3.4 Report Generation

3.4.1 Description

This feature allows the creation of annual statistical reports. Priority: **High**

3.4.2 Functional Requirements

- **REQ-FN-010:** The system must be able to generate a **Member Increase/Decrease Report** by year (**BM5.1**), showing statistics on the number of births, marriages, and deaths.
- **REQ-FN-011:** The system must be able to generate a **Member Achievement Report** by year (**BM5.2**), showing statistics on the quantity of each achievement type.
- **REQ-FN-012:** The user must be able to specify a time range (From year - To year) for the reports.

3.5 Rule Modification

3.5.1 Description

This feature allows the Administrator to customize the values in the available categories. Priority: **Medium**

3.5.2 Functional Requirements

- **REQ-FN-013:** The system must allow the Administrator to change (add/edit/delete) values in the categories: relationship type, hometown, occupation, according to **QD6**.
- **REQ-FN-014:** The system must allow the Administrator to change (add/edit/delete) values in the categories: cause of death, burial location, according to **QD6**.

4. Data Requirements

4.1 Logical Data Model

The system's data will be organized around a central Member entity. This model outlines the core entities, their primary attributes, and the relationships between them. This structure can be visualized as an Entity-Relationship Diagram (ERD).

Entity: Member

- **Description:** Represents an individual person in the family tree. This is the central entity to which most other data is linked.
- **Attributes:**
 - **MemberID** (Primary Key): Unique identifier for each member.
 - **FullName:** The member's full name.
 - **Gender:** The member's gender (e.g., Male, Female).
 - **DateOfBirth:** The member's date of birth.
 - **BirthPlaceID** (Foreign Key): Links to the Hometown entity.
 - **OccupationID** (Foreign Key): Links to the Occupation entity.
 - **Address:** The member's current residential address.
 - **IsDeceased:** A boolean flag indicating if the member has passed away.
 - **DateOfDeath:** The date the member passed away (nullable).
 - **CauseOfDeathID** (Foreign Key): Links to the CauseOfDeath entity (nullable).
 - **BurialPlaceID** (Foreign Key): Links to the BurialPlace entity (nullable).
 - **FatherID** (Foreign Key): A self-referencing key to another MemberID.
 - **MotherID** (Foreign Key): A self-referencing key to another MemberID.
- **Relationships:**
 - A **Member** can have one **Father** (who is also a Member).
 - A **Member** can have one **Mother** (who is also a Member).
 - A **Member** can be the parent to many other **Members**.
 - A **Member** can have one or more **Spouses**. This is managed through the **Marriage** entity.

Entity: Marriage

- **Description:** A linking table to manage the many-to-many relationship between members as spouses.
- **Attributes:**
 - **MarriageID** (Primary Key): Unique identifier for the marital relationship.
 - **Spouse1_ID** (Foreign Key): Links to the MemberID of the first spouse.
 - **Spouse2_ID** (Foreign Key): Links to the MemberID of the second spouse.
 - **DateOfMarriage:** The date the marriage occurred.

Entity: Achievement

- **Description:** Records notable accomplishments for each member.
- **Attributes:**
 - **AchievementID** (Primary Key): Unique identifier for the achievement record.
 - **MemberID** (Foreign Key): Links to the Member who earned the achievement.
 - **AchievementTypeID** (Foreign Key): Links to the AchievementType entity.
 - **DateOfAchievement:** The date the achievement was accomplished.

Lookup Table Entities: These tables store predefined options for various fields to ensure data consistency and support the "Change Rules" feature.

- **Hometown:** Stores the list of possible birthplaces (initially 4 options).
- **Occupation:** Stores the list of possible professions (initially 15 options).
- **CauseOfDeath:** Stores the list of possible causes of death (initially 12 options).
- **BurialPlace:** Stores the list of possible burial locations (initially 3 options).
- **AchievementType:** Stores the list of possible achievement types (initially 10 options).

4.2 Data Dictionary

Table: Member

Attribute Name	Data Type	Allow Null	Constraints / Validation Rules	Description
MemberID	INT (PK, AUTO_INCREMENT)	No	UNIQUE	Unique identifier for each member
FullName	VARCHAR(100)	No	NOT NULL	Full name of the member
Gender	ENUM('Male', 'Female')	No	NOT NULL	Gender of the member
DateOfBirth	DATE	No	CHECK(DateOfBirth ≤ CURRENT_DATE)	Member's birth date
BirthPlaceID	INT (FK → Hometown.HometownID)	No	FOREIGN KEY	Reference to hometown
OccupationID	INT (FK → Occupation.OccupationID)	No	FOREIGN KEY	Member's occupation
Address	VARCHAR(255)	Yes		Current living address
IsDeceased	BOOLEAN	No	DEFAULT = 0	Indicates if the member has passed away

DateOfDeath	DATE	Yes	CHECK(DateOfDeath > DateOfBirth)	Date of death (if deceased)
CauseOfDeathID	INT (FK → CauseOfDeath.CauseOfDeathID)	Yes	FOREIGN KEY	Reason of death
BurialPlaceID	INT (FK → BurialPlace.BurialPlaceID)	Yes	FOREIGN KEY	Burial location
FatherID	INT (FK → Member.MemberID)	Yes	CHECK(FatherID ≠ MemberID)	Reference to father
MotherID	INT (FK → Member.MemberID)	Yes	CHECK(MotherID ≠ MemberID)	Reference to mother

Table: Achievement

Attribute Name	Data Type	Allow Null	Constraints / Validation Rules	Description
AchievementID	INT (PK, AUTO_INCREMENT)	No	UNIQUE	Unique achievement ID
MemberID	INT (FK → Member.MemberID)	No	FOREIGN KEY	Member who achieved
AchievementTypeID	INT (FK → AchievementType.AchievementTypeID)	No	FOREIGN KEY	Type of achievement
DateOfAchievement	DATE	No	CHECK(DateOfAchievement ≥ DateOfBirth)	Achievement date validation

Table: Hometown

Attribute Name	Data Type	Allow Null	Constraints / Validation Rules	Description
HometownID	INT (PK)	No	UNIQUE	Unique hometown ID
HometownName	VARCHAR(100)	No	UNIQUE	Name of the hometown

Table: Occupation

Attribute Name	Data Type	Allow Null	Constraints / Validation Rules	Description
OccupationID	INT (PK)	No	UNIQUE	Unique occupation ID
OccupationName	VARCHAR(100)	No	UNIQUE	Occupation name (default 15 types)

Table: CauseOfDeath

Attribute Name	Data Type	Allow Null	Constraints / Validation Rules	Description
CauseOfDeathID	INT (PK)	No	UNIQUE	Unique ID for cause of death
CauseName	VARCHAR(150)	No	UNIQUE	Name of the cause of death (12 default types)

Table: BurialPlace

Attribute Name	Data Type	Allow Null	Constraints / Validation Rules	Description
BurialPlaceID	INT (PK)	No	UNIQUE	Unique burial place ID
PlaceName	VARCHAR(150)	No	UNIQUE	Burial location name (3 default places)

Table: AchievementType

Attribute Name	Data Type	Allow Null	Constraints / Validation Rules	Description
AchievementTypeID	INT (PK)	No	UNIQUE	Unique achievement type ID
TypeName	VARCHAR(150)	No	UNIQUE	Achievement type (10 default types)

4.3 Reports

This section specifies the content and parameters for the reports the system must generate.

4.3.1 Report 1: Member Growth and Decline Summary (BM5.1)

Description: This report provides an annual statistical summary of changes in the family's population over a specified period.

User Parameters:

- **Start Year:** The beginning year for the report range.
- **End Year:** The final year for the report range.

Content & Layout: The report will be a table with the following columns:

- **Year:** The calendar year.
- **Number of Births:** A count of all members whose DateOfBirth falls within that year.
- **Number of Marriages:** A count of all Marriage records where the DateOfMarriage falls within that year.
- **Number of Deaths:** A count of all members whose DateOfDeath falls within that year.

Example Output:

Year	Number of Births	Number of Marriages	Number of Deaths
2023	5	2	1
2024	3	4	2

4.3.2 Report 2: Member Achievements Summary (BM5.2)

Description: This report provides a summary of all achievements recorded across the family within a specified period, grouped by achievement type.

User Parameters:

- **Start Year:** The beginning year for the report range.
- **End Year:** The final year for the report range.

Content & Layout: The report will be a table with the following columns:

- **Achievement Type:** The name of the achievement (e.g., "Graduated University," "Published a Book").
- **Quantity:** A count of how many times that achievement was recorded for any member within the specified date range.

Example Output:

Achievement Type	Quantity
Graduated University	7
Community Service Award	3
Published a Book	1

4.4 Data Acquisition, Integrity, Retention, and Disposal

This section outlines the requirements for how the system will manage the data lifecycle, from creation to deletion, ensuring it remains accurate, available, and is handled properly.

4.4.1 Data Acquisition

Primary Method: All data will be acquired through manual entry by users assigned the Administrator role.

Interface: Data entry will be performed via secure web forms accessible after a successful login.

Initial Data Load: There is no requirement for an initial bulk data import. The family tree will be built incrementally by the administrator.

4.4.2 Data Integrity

The system must enforce data integrity at multiple levels to ensure the accuracy and consistency of the family tree information.

Input Validation:

- **Client-Side Validation:** Web forms must use client-side validation (e.g., JavaScript) to provide immediate feedback to users on data format errors (e.g., invalid date format, empty required fields) before submission.
- **Server-Side Validation:** All data submitted to the server must be re-validated on the server-side to protect against invalid or malicious data bypassing client-side checks.

Business Rule Enforcement: The application logic must enforce the following rules:

- A member's date of death must be chronologically after their date of birth.
- A relationship (e.g., "Child") must be linked to valid, existing parent members.
- An individual cannot be their own parent or spouse.

Database Constraints:

- **Referential Integrity:** Foreign key constraints must be used in the database to ensure that relationships cannot exist without valid corresponding member records.
- **Uniqueness:** A combination of Full Name and Date of Birth should be unique to prevent duplicate member entries.
- **Required Fields:** Essential fields, such as a member's name and date of birth, must be defined as NOT NULL at the database level.

4.4.3 Data Retention and Backup

Data Retention Policy:

- Active user data will be retained indefinitely as long as the system is in operation.
- Data related to deleted members will be handled according to the Data Disposal policy (see section 4.4.4).

Data Backup and Recovery: A robust backup and recovery strategy is required to protect against data loss due to hardware failure, software bugs, or accidental deletion.

- **Backup Frequency:** The database must be backed up automatically on a daily basis.
- **Backup Storage:** Backups must be stored in a secure, geographically separate location (off-site) from the primary web server to protect against physical disasters.
- **Backup Retention:** Daily backups will be retained for a rolling period of at least 30 days.
- **Recovery Plan:** A documented Data Recovery Plan (DRP) must be in place, outlining the procedures to restore the system from a backup. The plan should target:
- **Recovery Point Objective (RPO):** No more than 24 hours of data loss.
- **Recovery Time Objective (RTO):** The system should be fully operational within 4 hours of initiating the recovery process.

4.4.4 Data Disposal

Deletion Mechanism: The Administrator role will have the sole permission to delete member records from the family tree.

Deletion Method:

- A soft-delete mechanism is required. When an administrator deletes a member, the record will be marked as "inactive" in the database but not permanently erased. This allows for the possibility of reversal in case of accidental deletion.
- Soft-deleted records will not appear in the public family tree view, searches, or reports.

Cascading Effects: The system must define how to handle records linked to a soft-deleted member. For instance, relationships involving the deleted member should also be considered inactive.

Permanent Deletion: There is no requirement for a user-facing permanent deletion feature. The permanent removal of soft-deleted data will be managed at the database level by the system maintainers after a defined period (e.g., 90 days), or upon special request.

5. External Interface Requirements

This section provides information to ensure that the system will communicate properly with users and with external hardware or software elements.

5.1 User Interfaces

- The system must have a user-friendly web-based Graphical User Interface (GUI).
- The interface language is English.
- The layout of the main data entry pages should reference forms BM1, BM2, and BM4.

5.2 Communications Interfaces

The system will communicate with the user's browser via the **HTTPS** protocol to ensure that data transmission is encrypted and secure.

6. Quality Attributes

6.1 Usability

The website must be easy to learn and use for non-technical users.

6.2 Performance

- Average page load time should not exceed 3 seconds.

- Response time for search and report generation operations must be under 5 seconds.

6.3 Security

- **Authentication:** The system must have a username and password login mechanism for all users.
- **Authorization:** There must be at least two user roles (Administrator and Viewer) with different access rights to prevent unauthorized data modification.
- **Encryption:** All traffic between the client and server must be encrypted using HTTPS.
- **Protection:** Measures must be in place to protect the website from common web security vulnerabilities

6.4 Safety

The biggest risk is data loss. The server must have a periodic database backup mechanism to ensure restorability in case of an incident.

7. Other Requirements

- **Deployment:** The website must be deployed on a reliable hosting service.
- **Domain Name:** A suitable domain name must be registered for users to access the site.
- **Maintenance:** A regular maintenance plan for the server and database is required.

Appendix A: Glossary

Family Tree: A chart or record that traces the history of generations in a family line.

HTTPS: Hypertext Transfer Protocol Secure.

Appendix B: Analysis Models

The forms (BM1, BM2, BM3, BM4, BM5.1, BM5.2) in the "Project List" document are considered prototypes for the user interfaces of the corresponding features.