Software Requirements Specification

for

KinTree

Version 1.0 approved

Prepared by
Owen Adams, Kennedi James, Destiny Milsap, and Jade Thompson

Mississippi State University
Software Engineering Senior Project

October 6, 2024

1. Introduction

1.1 Purpose

This software requirements specification (SRS) document describes the initial version of KinTree (1.0). The following document will detail the application's intended scope, major functionalities, system features, and nonfunctional requirements. The application described by this SRS is a web application that will serve as a social platform for building and sharing personalized family trees.

1.2 Document Conventions

System features and nonfunctional requirements are listed in descending order of importance to the overall performance of KinTree. Sub-requirements hold equal importance to their respective high-level requirements.

1.3 Intended Audience and Reading

This document is intended for use by the developers and clients of the KinTree project. The following sections provide further detail on the application's scope, use cases, system features, and other requirements.

All users of this document should familiarize themselves with the purpose, intended usage, and overall description of KinTree, which can be found in Section 1 and 2 of this document. Users who wish to know more about the granular features and specifications of the application, such as developers, should read the remainder of the sections within this SRS.

1.4 Product Scope

In developing KinTree, we aim to create a platform that allows users to dynamically create and share personalized family trees. Users of KinTree will be able to seamlessly import known information about their immediate and extended family, and then view a visualized representation of these familial connections. KinTree will also serve as a platform for storing related media, such as pictures and videos, of family members. This will allow users to generate and preserve a comprehensive record of their family heritage and memories, ensuring that valuable stories and connections are not lost over time.

Our team will also foster collaboration within the KinTree application. Users will be able to share pictures and videos with their family members, which will promote widespread sharing of memories and experiences. Additionally, to allow users to expand their recorded family trees beyond their personal knowledge, KinTree will allow users to share family trees with each other. Users will have the option to view shared family trees, and merge mutual family members from other trees into their own personal trees. Enabling such collaboration will contribute to a dynamic and growth-oriented application, which will benefit users and other stakeholders alike.

1.5 References

[1] React Official Website
https://react.dev/

[2] Node.js Official Website
https://nodejs.org/en

[3] MySQL Official Website
https://www.mysql.com/

[4] NIST Usability Standards
https://www.nist.gov/programs-projects/usability-standards

2. Overall Description

2.1 Product Perspective

Kintree is a collaborative website application that aims to provide its users with the ability to create, share, and manage their family trees. These trees, once shared, can be transformed into personalized relation trees based on the recipient's relation to the sender. A user will have their base tree that remains on their end of Kintree. A user will be able to share their tree with other family members, the recipient can edit the shared tree and send it back, and the user can save the shared tree with changes as their base tree if desired. The website integrates with iOS and Android contacts in a cross-platform fashion, which allows users to seamlessly upload their family members to the tree, maintaining an up-to-date and accurate version of the family tree. Some key features of this application include dynamic relation tree generation, memory sharing through photo and video uploads to the website, and built-in notifications to remind users of important dates and facilitate communication. Kintree emphasizes user privacy, offering customizable access permissions and data handling.

2.2 Product Functions

- A user will be able to login, edit their base family tree, save their shared tree as their base tree, respond to and send notifications through a chat feature to other family members, view information about each family member in a tree, and logout.
- An admin user will be able to login, oversee users' activity, approve/deny account creation requests, and logout.

2.3 User Classes and Characteristics

- User: The user will be able to login, edit their base tree information, view family member information on their tree, share a tree with 10 other family members, merge a received shared tree with their own base tree, and logout.
- Admin: The admin user can login, view account info for registered users, approve new account creation requests from users, deny new account creation requests from users, and logout.

2.4 Operating Environment

The application will primarily run on Windows OS and will be cross-platform, making it capable of running on iOS and Android mobile devices. It will be a web-based app that uses a framework consisting of React [1] for front and middle-tier, as well as Node.js [2] and MySQL [3] for backend/server-side.

2.5 Design and Implementation Constraints

React is extremely well-suited for the creation of websites, but it also poses some challenges. One of which is the fact that it is not a fully-fledged framework, since it mainly handles the view part of the model-view-controller architecture for web apps. Frameworks like Angular have more capabilities for modelling and controlling as well as viewing. Other libraries will need to be accessed to handle the things that React will not be able to create, so it will be challenging to use React to its fullest extent.

Another thing about React is that it commonly brings code bloat with it, which means that to run even minor functionalities, massive amounts of code and libraries will need to be accessed that are often external to React itself. This could lead to slow loading times for the web app, especially if the network has weak signal strength.

One issue with MySQL for website apps is that it may have issues handling storage of large amounts of data. MySQL does not have support for the storage of large loads of read/write data. These issues are near definite only whenever MySQL is used to handle millions of records and transactions however, so if the website can be smaller, then it will be better off.

3. System Features

3.1 User Registration and Authentication

3.1.1 Description and Priority

This feature allows new users to create accounts and existing users to securely login. New users can create accounts using their emails. The system should validate and securely store user credentials. Registered users can log in securely using their registered credentials. The systems should include authentication, including password checks. This is of high priority because it allows users' information to be logged.

3.1.2 Functional Requirements

- The system shall provide a user registration form with fields for password and username, email, phone number, or social media (OAuth integration with Google, Facebook, etc.). There are also fields for optional personal information.
- The system shall validate the email format and ensure it is unique. If incorrectly formatted or not unique, the system will display an error message.

- The system shall enforce password strength requirements. If the requirements are not met, the system will display an error message.
- Upon successful registration, the system shall securely store the user's credentials and create a new user account.
- The system shall provide a login page with the field for username/email and password.
- The system shall validate user's credentials by checking against the stored user data. If the user's credentials are invalid, the system will display an error message.
- The system shall provide an option for users to reset their passwords in case they have forgotten the password. This option could be in the form of a "Forgot Password" link.
- Upon clicking the "Forgot Password" link, the system shall provide a password recovery page where they can insert their email.
- The system shall send a unique password reset link that expires after a specific time.
- The system should validate for password strength during the reset process and update input if valid.
- The system shall support password hashing and salting for secure storage of user passwords
- Secure login with two-factor authentication.
- The system shall provide users with a way to customize their profile information (name, photo, etc.).

3.2 User Activity Dashboard

3.2.1 Description and Priority

This feature gives users the ability to view an activity dashboard. Medium priority.

3.2.2 Functional Requirements

- The system shall give users a dashboard that provides an overview of recent family activities (e.g. added members, new memories, upcoming events).

3.3 Add a New Family Member

3.3.1 Description and Priority

This feature allows users to add a new family member to their tree. This is of high priority.

3.3.2 Functional Requirements

- The system shall provide the user with the option to add a new member.
- The system shall provide the user with a form to add required information with fields for first name, last name, and date and birthday.
- The system shall provide the user with the option to add information about the member such as location, birthplace, death date etc.
- The system shall allow the user to identify the familial relationship to the user.
- The system shall validate the new member and provide the user with a summary of the information inputted.

- Upon new member validation, if the user with identical required information, the system shall notify the user.
- The system shall provide the user with the option to create a new member or update the existing member.

3.4 Family Tree Creation and Management

3.4.1 Description and Priority

This feature allows the user to create and manage a base family tree. High priority.

3.4.2 Functional Requirements

- Users should be able to create a base family tree by adding individuals with personal details (e.g., password reset, two-factor authentication).
- The system must support dynamic relation tree generation, allowing users to share their tree and adjust the view based on the recipient's relation.
- Users should be able to edit, remove, or update individuals and relationship types on their tree.
- There must be an option to import/export family trees in a common format like GEDCOM.

3.5 View Family Tree

3.5.1 Description and Priority

This feature allows the user to see their family members in a hierarchical manner.

3.5.2 Functional Requirements

- The system shall provide the user with a visualization of the members of their family including their names.
- The members of the family should be able to be view from user -> parents -> grandparents -> etc.
- The system shall allow the user to click on the family member to see additional information.

3.6 Edit Existing Family Member

3.6.1 Description and Priority

This feature allows users to edit the information of an existing family. This is of high priority.

3.6.2 Functional Requirements

- The system shall provide the user with the option to edit a family member's information.
- The system shall provide the user with a form to edit existing information.
- The system shall provide the user with the ability to fill in empty optional categories.
- The system shall provide the user with an option to back out of updating the existing information.

3.7 Search and Filtering

3.7.1 Description and Priority

This feature gives the users the ability to search for family members or media within a tree. Medium priority.

3.7.2 Functional Requirements

- The system shall allow users to search for family members or media within the tree by name, date, or tag.
- Filters should be available to view specific branches of the tree or narrow down memories by time, event, or person.

3.8 Share Family Tree

3.8.1 Description and Priority

This feature allows users to share their current family tree with family members. This is of high priority.

3.8.2 Functional Requirements

- The system shall provide the user with the option to send their current base family tree to another person.
- The system shall provide the user with the options to share via email, text, etc.
- The system shall provide the user with a list of questions in relation to the person they are sending it to.

3.9 View Shared Trees

3.9.1 Description and Priority

This feature allows a new user to view a shared family tree from an existing user. This is of high priority.

3.9.2 Functional Requirements

- The system shall provide the user with a clickable link to a page where they can view the existing tree.
- The system shall direct the user to the sign up/ in page for any further actions.
- The system shall provide the user with a list of questions to verify the relation to the existing user.
- The system shall provide the user with an option to update the existing tree for personal use.
- The system shall provide the user with the option to simply view the tree or merge it with their base tree.

3.10 Privacy and Access Control

3.10.1 Description and Priority

This feature allows the user to establish privacy and viewing settings on their family tree. High priority.

3.10.2 Functional Requirements

- The system shall allow users to define privacy settings, controlling who can view, edit, or comment on their family tree and shared memories.
- The system shall allow users to restrict access to specific parts of the tree (e.g. certain branches, sensitive information).
- The system shall allow the user to implement different permission levels (view-only, editor, admin) for family members.
- The system shall allow users to revoke or modify access rights at any time.

3.11 Merge/Set Family Tree from Shared Tree

3.11.1 Description and Priority

This feature allows a new user to update their base family tree based on a shared family tree from a family member. This is of high priority.

3.11.2 Functional Requirements

- The system shall provide the user with the option to identify the relationship of the sharing family member.
- The system shall provide the user with a list of possible family members.
- The system shall allow the user to choose the members (selectively merge) that would be in their tree.
- The system shall provide the user with the option to define the new family member's relation to them.
- The system shall update the view of the user who is receiving the shared tree so that their relation to the sharer and their family is updated (e.g. nephew shares with an uncle).
- The system shall provide the user with a comparison view of the received shared tree and their base tree to show differences in each.
- After selecting people to merge to base tree, the system shall integrate the selected people into the user's base tree, adjusting relationships accordingly.

3.12 Upload Media to Family Tree/ Memory Sharing

3.12.1 Description and Priority

This feature allows the user to upload photos, videos, voice memos, and other media to a family member's profile. This is of medium priority.

3.12.2 Functional Requirements

- The system shall provide the user with an option on the family members' information profile to add media.
- The system shall allow different types of media to be uploaded.
- The system shall allow the user to add a defined number of photos, videos, etc.
- The system shall allow the user to add metadata such as dates, event names, and other information.
- The system shall provide a memory gallery with media that family members can scroll through, comment on, and interact with.
- The system shall provide the user with the ability to delete or edit uploaded memories.

3.13 Backup and Restore

3.13.1 Description and Priority

This feature gives users the ability to restore previous trees.

3.13.2 Functional Requirements

- The system shall give users the option to restore their family tree and data from previous backup points.

3.14 Data Export and Download

3.14.1 Description and Priority

This feature allows users to move trees and memories to new formats. Low priority.

3.14.2 Functional Requirements

- The system shall give users the ability to export their entire family tree and memories into common formats (GEDCOM, ZIP for media files).
- The system shall provide users with options for downloading family photos or stories individually or in bulk.

3.15 Generational Archiving and History Preservation

3.15.1 Description and Priority

This feature allows users to archive and give access to historical trees. This is low priority.

3.15.2 Functional Requirements

- The system shall allow the user to archive family trees for future generations, preserving historical records.
- Users can assign a successor (family member) who will inherit responsibility for maintaining and updating the tree.
- Older trees should remain accessible for viewing by authorized family members.

3.16 Chat with Family Members

3.16.1 Description and Priority

This feature allows users who are related to chat with one another. This is of low priority.

3.16.2 Functional Requirements

- The system shall provide users with a separate page where direct messages can be sent to one another.
- The system shall allow the users to pick from a list of family members that use the application to message.
- The system shall log information such as the time and date of the message.
- The system shall store the message between the users until it is deleted from a user's account.
- The system shall support real-time messaging for quick family communications.
- Users shall receive notifications for unread messages or ongoing discussions.

3.17 Visualize Family Map

3.17.1 Description and Priority

This feature allows users to visualize the locations of their family members on a map. This is low priority.

3.17.2 Functional Requirements

- The system shall provide users with a separate page to view a map.
- The systems shall allow family members' locations to be viewed on a map represented by pins.
- The system shall allow the user to zoom in and out on the map.

3.18 Sync Contacts

3.18.1 Description and Priority

This feature allows users to sync their existing contacts. This is of low priority.

3.18.2 Functional Requirements

- The system shall provide users with an option to sync their contacts while using their mobile devices (iPhone and Android).
- If the users deny the option, the system shall provide a message verifying the options.
- If the user accepts the option, their contacts shall be allowed to sync.
- The system shall allow the user to pick the family members and add the information to the correct information profile.
- Updates to contacts in the user’s phone should reflect in the tree.

3.19 Family Event Planning

3.19.1 Description and Priority

This feature allows the user to create events and share event details with family members invited. This is of medium priority.

3.19.2 Functional Requirements

- The system shall allow users to create family events (e.g. reunions, weddings) and invite family members.
- The system should track RSVPs and provide notifications about event updates.
- Users should be able to share event details, locations, and times with invited members.

3.20 Important Event Message Board

3.20.1 Description and Priority

This feature allows the user to see important information regarding family members. This is of low priority.

3.20.2 Functional Requirements

- The system shall provide users with a separate page to view the reminders/ announcements.
- The system shall allow users to add and manage reminders for these events.
- The system shall give a notification upon logging in on days of important dates.
- The system shall add a list of important dates such as birthdays, anniversaries, etc.
- The system shall give the user the ability to opt out/in for receiving notifications via email, SMS, or app notifications.

3.21 Audit and Activity Logging

3.21.1 Descriptions and Priority

This feature allows users to access tree version history. Medium priority.

3.21.2 Functional Requirements

- The system shall allow users to access version history to review or revert changes to their tree or memories.
- The system should log changes made to the family tree (who edited what and when).

3.22 Multilingual Support

3.22.1 Description and Priority

This feature ensures that the website caters to users with differing languages. High priority.

3.22.2 Functional Requirements

- The system shall support multiple languages to cater to users from different regions.
- Family members shall be able to interact with the tree and system in their preferred language.

3.23 Mobile App Integration

3.23.1 Description and Priority

This feature adds the option for users to have a mobile app version of the web application. Medium priority.

3.23.2 Functional Requirements

- The system shall allow users to have companion mobile apps on iOS and Android.
- The system shall allow users to perform all core functions (tree creation, memory sharing, messaging) from the mobile app.
- Push notifications should be available on mobile for reminders, events, and communications.

3.24 User Feedback and Support

3.24.1 Description and Priority

This feature allows users to send feedback and access FAQs through the website. High priority.

3.24.1 Functional Requirements

- The system shall allow users to report issues or submit feedback directly from the website/app.
- The system shall provide users with access to FAQs and user support options (live chat, email, etc.).

3.25 Approve/Deny Accounts

3.25.1 Description and Priority

This feature allows the admin to approve/deny accounts. This is of medium priority.

3.25.2 Functional Requirements

- The admin shall have an accessible page on the website with new account creation requests.

- The system shall allow admin to click on each request and be able to see the information the user put in during registration of account including name, email, address, ID, and validate the account in their own discretion.
- The system shall allow the admin to approve the creation of account by clicking the "Approve" button on the page.
- The system shall allow the admin to deny the creation of account by clicking the "Reject" button on the page

4. Other Nonfunctional Requirements

4.1 Performance Requirements

4.1.1 Response Time:
The system should respond to user actions, such as loading family trees or searching for members, within 2 seconds under typical conditions (up to 100 users). For heavier loads (up to 1000 users), responses should not exceed 5 seconds.

4.1.2 Data Processing:
Family trees and shared media should update in real-time with minimal delays. Uploads for media files under 100MB should take no longer than 10 seconds on a stable connection.

4.2 Safety Requirements

4.2.1 Data Protection:
User data (family trees, media) must remain intact in the event of a system failure. Daily automatic backups should ensure data recovery if needed.

4.2.2 Error Handling:
Unexpected errors, like network issues, should be managed with clear notifications to users, and their progress should be auto saved to prevent data loss.

4.3 Security Requirements

4.3.1 User Authentication:
All users must log in with a unique username and password. Passwords should be securely hashed using standard encryption methods.

4.3.2 Data Encryption:
Sensitive user information, including personal details and media, must be encrypted both during transmission and when stored on the servers.

4.3.3 Audit Trail:

The system must log significant actions, such as logins and edits, for security purposes.

4.4 Software Quality Attributes

4.4.1 Usability:
KinTree's interface should be user-friendly and intuitive for people of all ages, with simple navigation and optional tutorials to guide users through key features.

4.4.2 Reliability:
The application should run smoothly with minimal downtime and be able to function across a range of internet connection speeds.

4.4.3 Maintainability:
The system should be easy to update and maintain, with clear documentation for developers to manage future enhancements or bug fixes.

5. Other Requirements

KinTree should abide by all applicable privacy laws and legislation, especially considering the sensitive nature of information shared within the application. Efforts should be made from a development standpoint to prioritize privacy and security in all aspects of the application, including protections in the client-side application and server/database. Cyber-attacks should be actively prevented and protected against during development and throughout efforts to sustain the application.

The application should also comply with commonly accepted accessibility and usability standards [4] to maintain a widened potential audience of end users. Given the multi-generational nature of KinTree's intended user population, younger and older users alike should be able to easily navigate the application's interface and fully make use of provided features.

Appendix A: Glossary

React: a JavaScript library for building web interfaces that allows for component-based UI design

Node.js: a cross-platform asynchronous JavaScript runtime environment

JavaScript: a multi-purpose language commonly used in web development applications for client-side and server-side scripting

MySQL: a relational database management system used for storing various forms of data