# Software Requirements Specification

**for**

# Roots

**Version 1.0 approved**

**Prepared by Sarah Colletta, William Turner, & Tom Rigney**

**SUNY Oswego Software Design**

**16-Feb-2016**

**Software Liscense**

Roots is a program designed to collect data on users lineage and display this information to the user in a family tree and parliament diagram.


**Website Liscense**

# Table of Contents

# Revision History

| Name | Date | Reason For Changes | Version |
|---|---|---|---|
| | | | |
| | | | |

# 1. Introduction

This section provides a scope description, product purpose and overview of information included in this SRS document.

## 1.1 Purpose

The purpose of this document is to present a detailed description of the Roots- a web-based genealogy tracing system. It will illustrate the features of the full system, the interfaces of the system, system constraints and interactions between the system and other external applications.This is a web-based system used to collect and display family tree data in a user friendly manner.

## 1.2 Intended Audience and Reading Suggestions

This document is intended for all individuals participating in and/or supervising the Roots project which includes both the stakeholders and the developers of the system. Readers interested in a brief overview should focus on the rest of Part 1 (Introduction) as well as Part 2 (Overall Description).

Readers interested in exploring the features of Roots in more detail should focus on Part 3 (External Interface Requirements) which further technical details including information on the user interface, and hardware and software platforms the system will run on, and Part 4 (System Features) of this document which expands upon the information laid out in the Overall Description.

Readers interested in non-technical aspects of the project should focus on Part 5 (Non-functional Requirements) which covers performance, security and *various other attributes that will be important to users.*

## 1.3 Product Scope

Roots is a Genealogy Tracing System for individuals seeking to discover their family history. This system is designed to allow users to create their family tree and “grow” it by adding family members. Through the creation of individual family trees, the system will meet the user’s desire to learn more about their family members and family history while remaining easy to understand and use. This is a web-based application which is accessible across multiple Internet browsers including Chrome, FireFox, Safari, and Internet Explorer. There is no cost to sign up for this service.

### 1.4 References

- Ancestry.com
- geni.com

# 2. Overall Description

## 2.1 Product Perspective

Currently, several genealogy tracing websites already exist, we are only creating this one so that we can pass a core class and get credit to graduate.

Roots is a genealogy website which allows the user to build a family tree to track their individual family history, starting with themselves. Users first create an individual profile upon registration which includes some basic as well as more detailed information about themselves. They can then begin creating their own family tree by adding family members and including any known information about those individuals, and view information pulled from our database. We aim to provide a genealogy tracing platform that is appealing and user friendly.

The scope of the project encompasses both server and client-side functionalities. Both aspects are covered in detail within this document.

## 2.2 Product Functions (use cases)

- Allow users to search for family members
- Allow users to view their family tree
- Allow users to add family members to their family tree
- Allow users to edit/delete family members on their family tree

## 2.3 User Classes and Characteristics

The following is a list of primary, secondary and tertiary users and the features they will be allowed to use.

- Primary user: Individuals actively seeking family history, a member of the site who has permission to add, edit and delete individuals.
- Secondary user: Non-members who are granted limited access. These users are only allowed access to the Home, About and Contact pages.
- Tertiary user: An individual who is not an active member of the site, nor a visitor but whose information is on the site and is affected by the use of the site by others.

## 2.4 Operating Environment

The software will operate on all platforms and operating systems(- iOS, Windows, etc.) and across web browsers including Chrome, Safari, Internet Explorer and Firefox. Account users' computers will require 1GB RAM for the install program

## 2.5 User Documentation

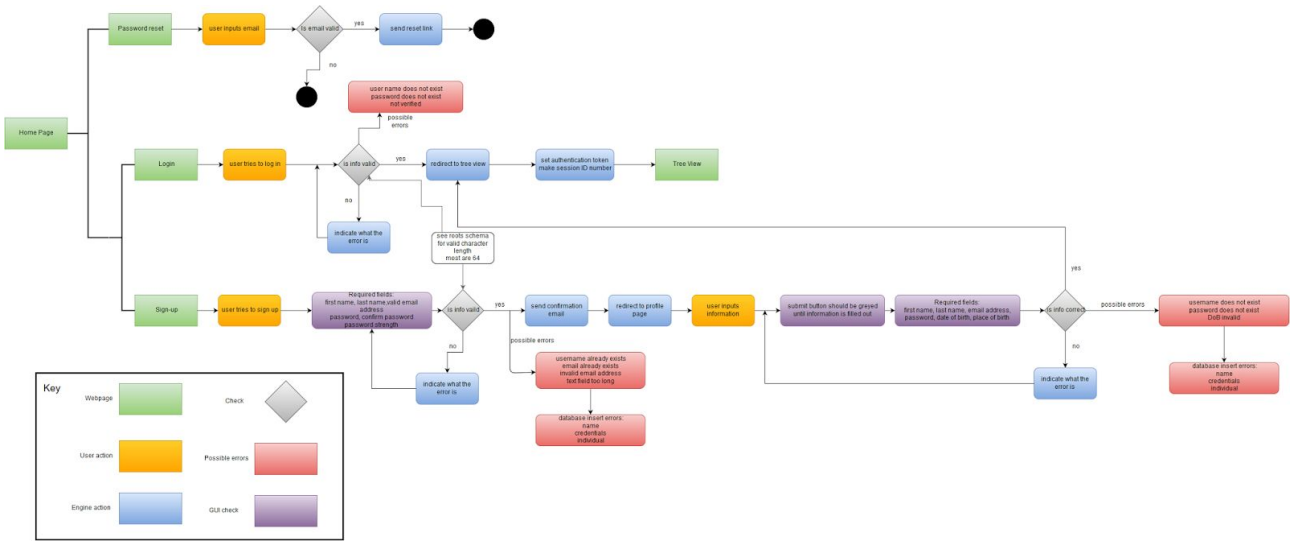
Should a user experience any issues, a user manual is available for viewing on the website for help documentation.

## 2.6 Assumptions and Dependencies

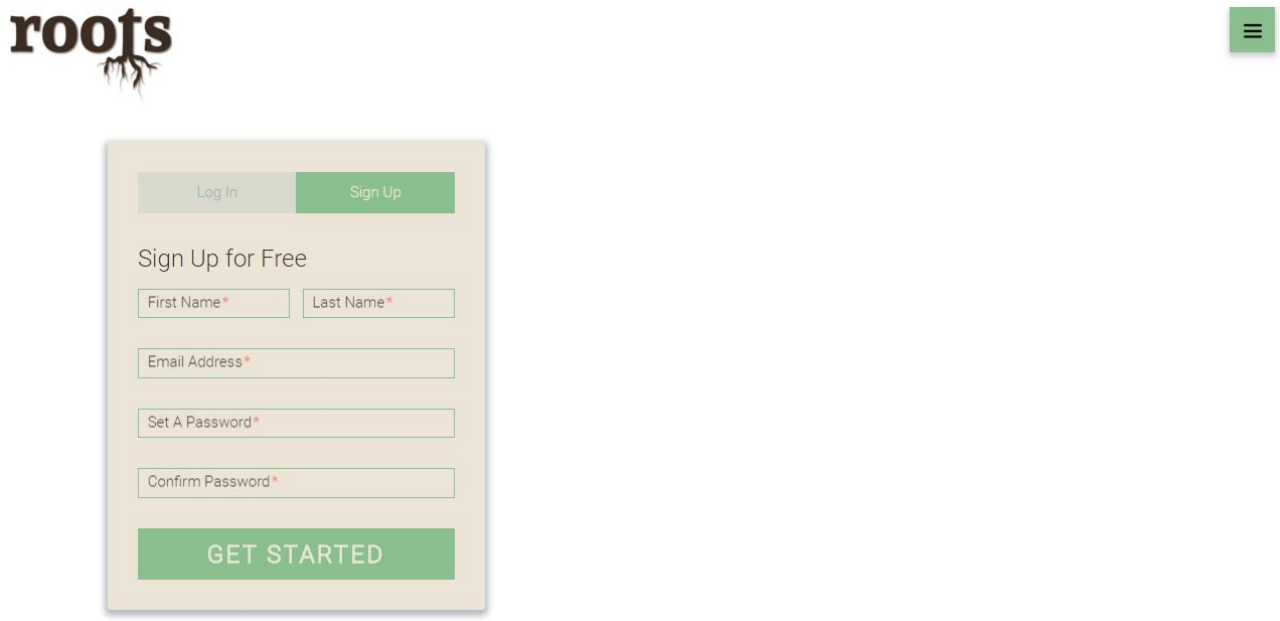
One assumption about this product is that it will be primarily used in desktop format and that users will not be adding several text fields on their mobile devices. However, the interface has been made responsive, meaning that all elements of the web page re-size to fit mobile screen sizes.

# 3. External Interface Requirements

## 3.1 User Interfaces

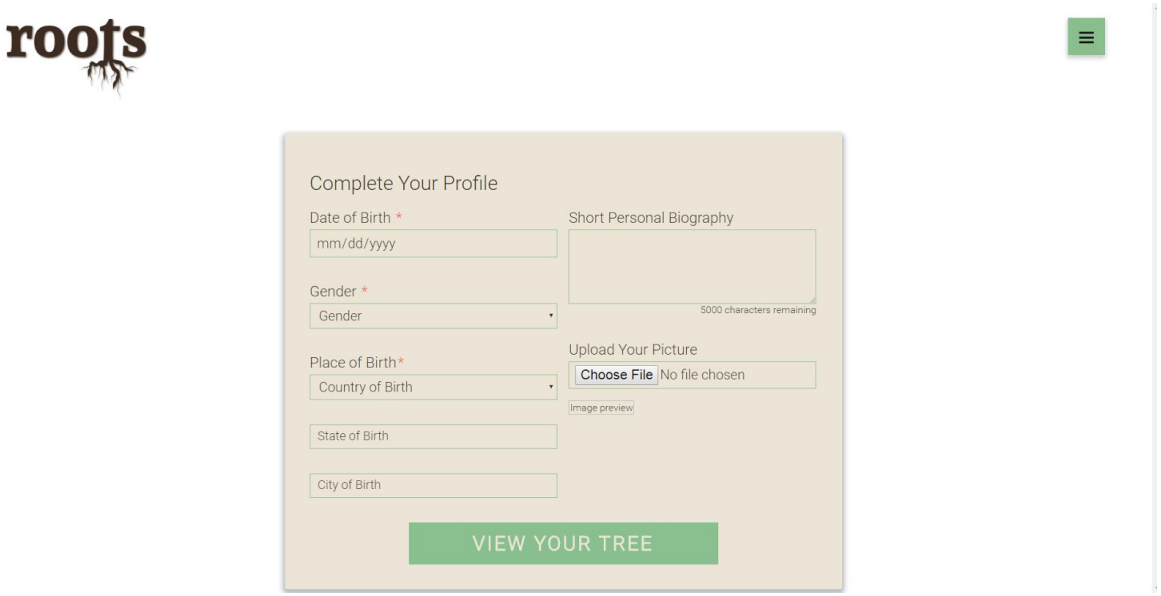


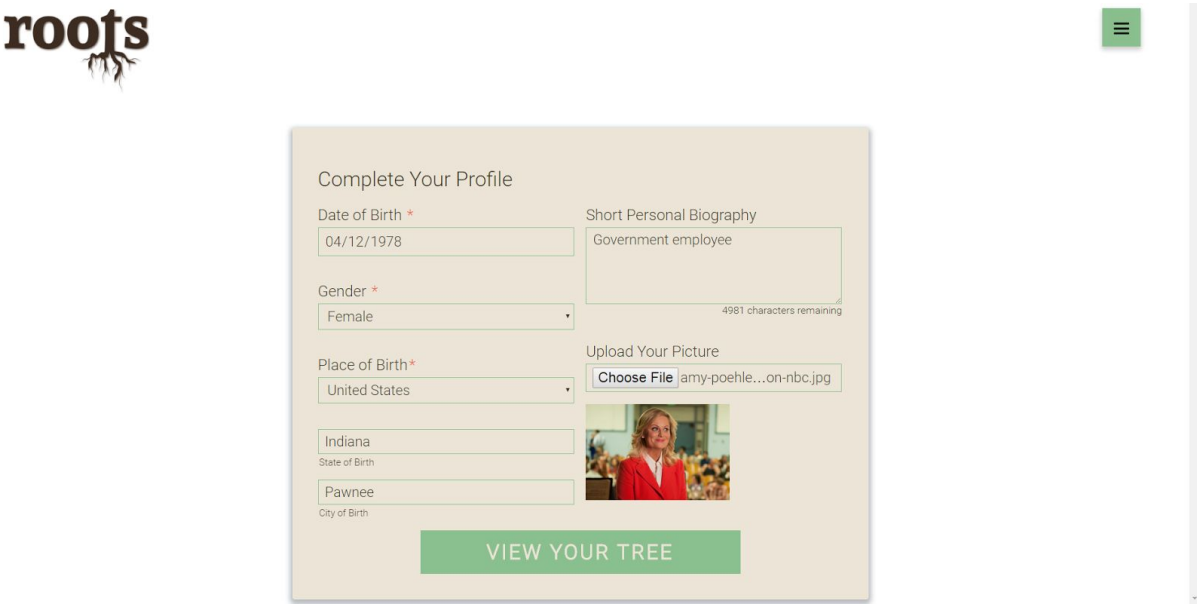
**Log in / Sign up page**




- For first time and returning users.
- Prompts the user to either sign up for Roots or enter their log in information.
- New users are required to provide their first and last name, email and create a password which must be entered twice (once for verification).
- Contact and About links are present on all pages

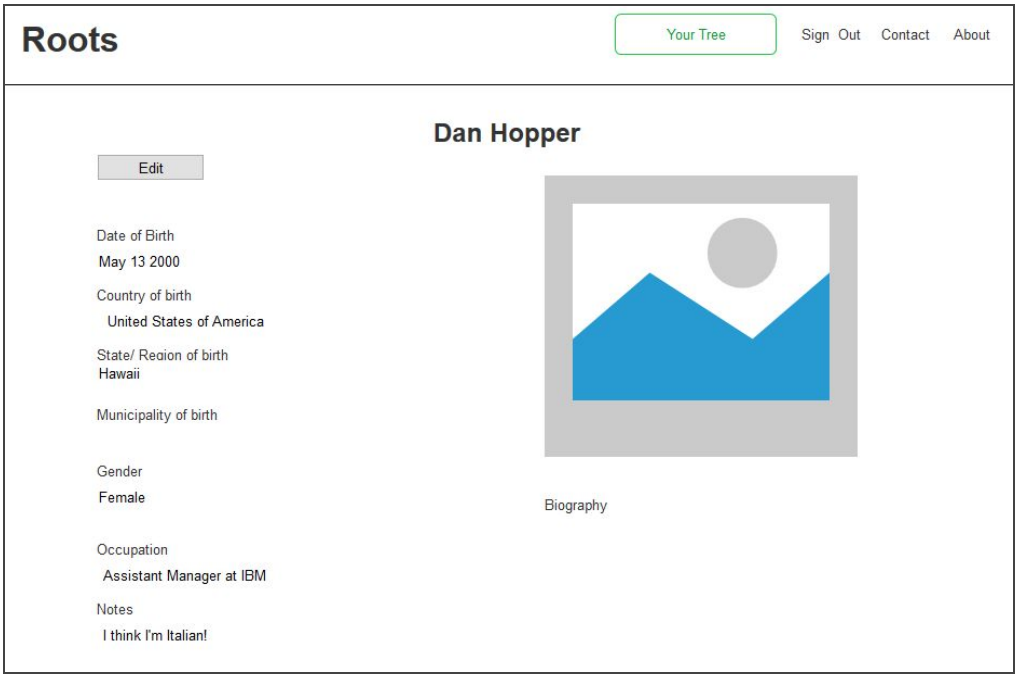
**Profile page**





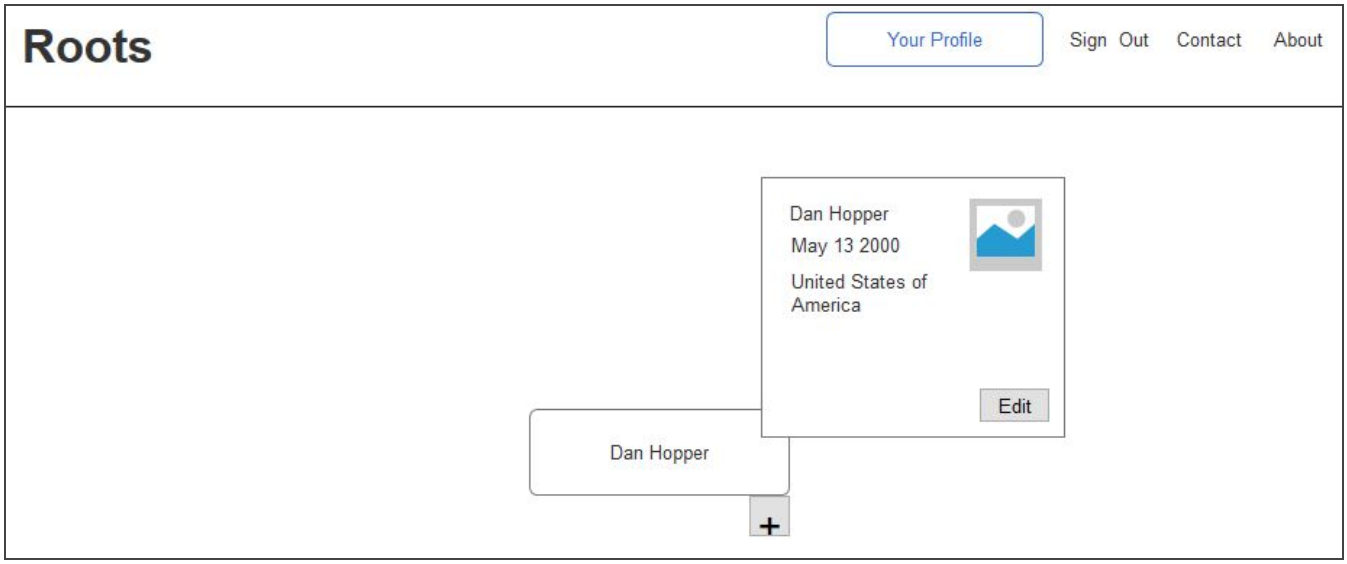

- For new user sign up
- Drop down menu for the following fields: Date of birth (month, day, year), Gender, Citizenship at birth, and State of birth.
- Occupation, Notes and Biography fields are open text fields in which the user can freely enter information.
- Contact and About links are present on this page, additionally, user may sign out.

**Completed profile**



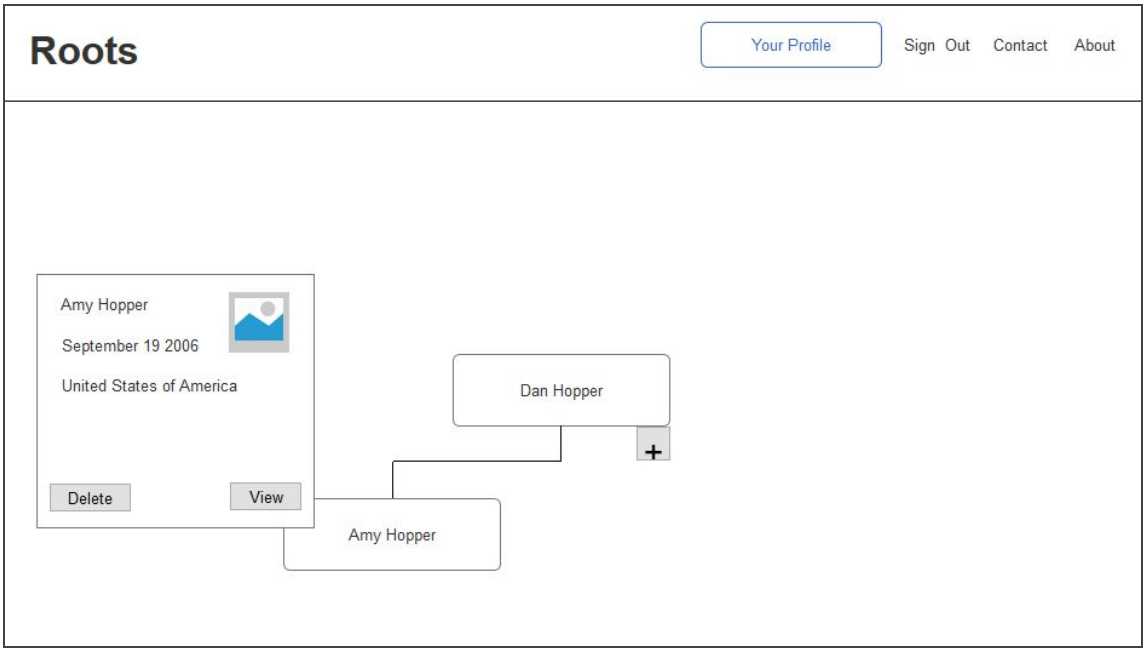

- Shows the information entered from profile creation
- User may edit this information when they please
- Contact and About links are present on all pages, additionally, user may sign out or view their family tree from this page

**Tree page**



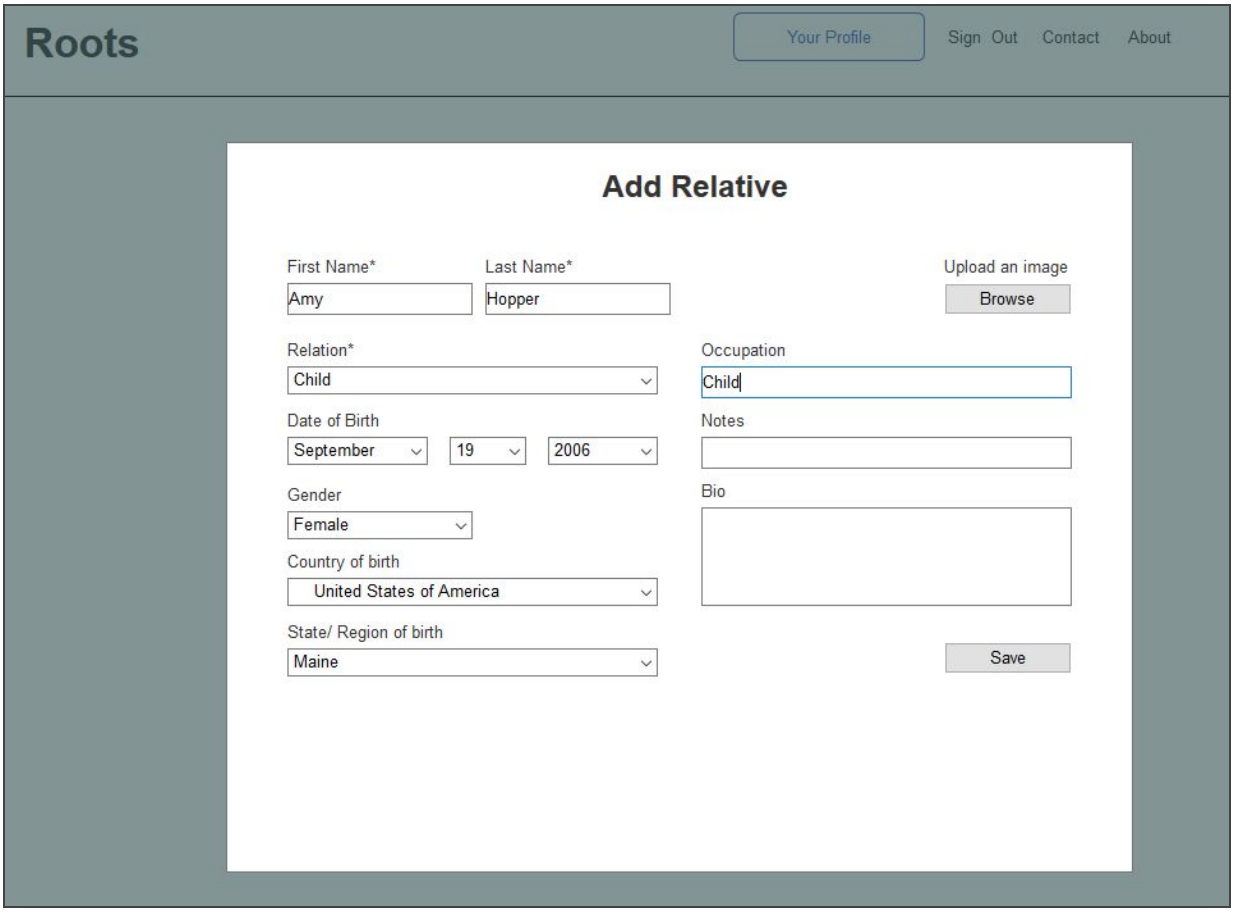

- Shows (graphically) the user's family tree with the branches extending from the user
- User may edit this information when they please
- Contact and About links are present on all pages, additionally, user may view their profile from this page

**Update additions to tree**




- User may add members to their family tree from this screen
- User hovers over their name and a plus symbol appears, once it is clicked the user will be prompted to enter information about relative

**Add relative**




- User accesses “Add Relative” in a modal window.
- User must add the following information on their relative: Name, Date of birth, Date of death, Place of birth, Place of death, relation to user.

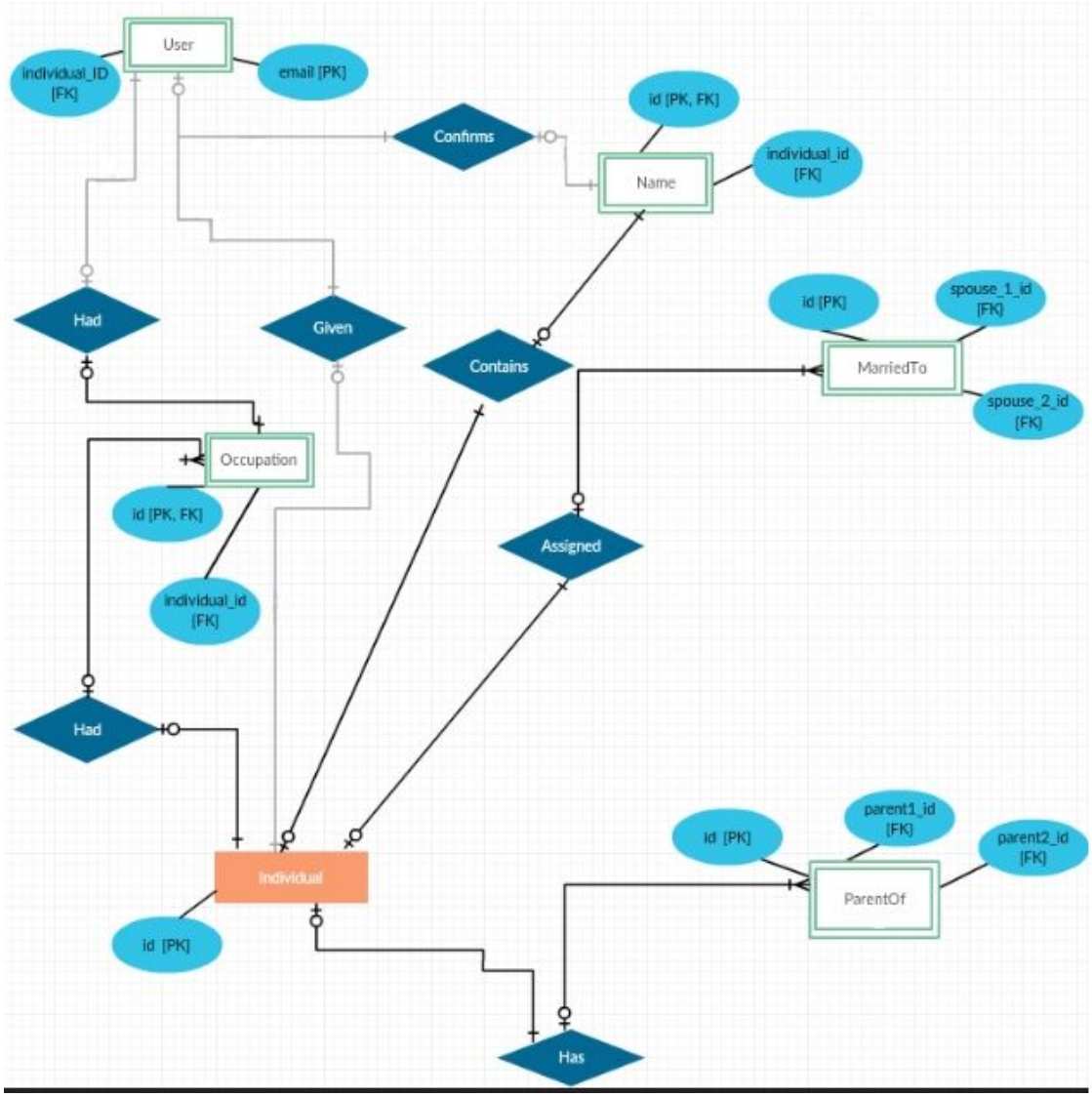
**Style Sheet**

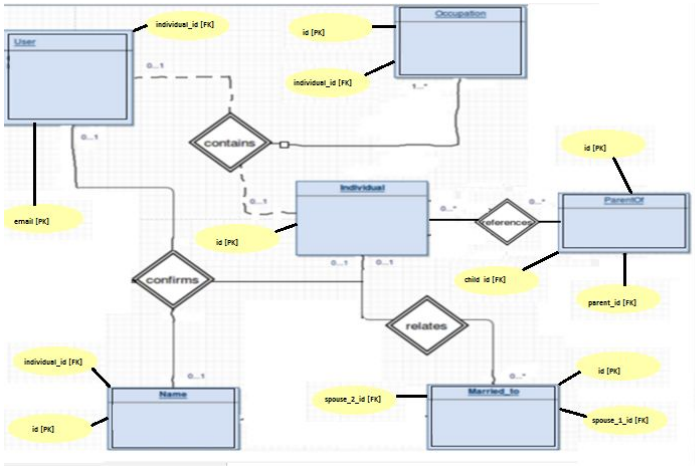

## 3.2 Software Interfaces

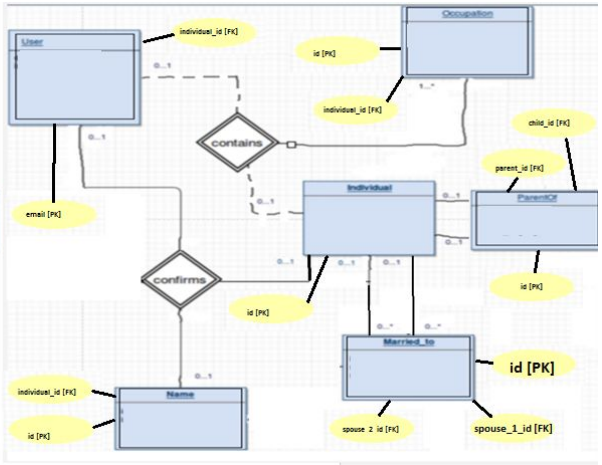
The Roots web app is to be developed using node.js, IntelliJ IDEA, sails, javascript, HTML and CSS.

3.2.1 Incoming and Outgoing Items
Outgoing data consists of family tree, data on individuals including name, date of birth, place of birth, date of death, occupation, etc.
this information is sent by users to the server.

Incoming data consists of updates from the server regarding member family/ relative information, as well as any notifications deemed necessary.


User
individual_ID [FK]
email [PK]
Confirms
id [PK, FK]
Name
individual_id [FK]
Had
Given
Contains
id [PK]
spouse_1_id [FK]
MarriedTo
spouse_2_id [FK]
Occupation
id [PK, FK]
Assigned
individual_id [FK]
Had
Individual
id [PK]
parent1_id [FK]
parent2_id [FK]
ParentOf
id [PK]
Has


User
Occupation
contains
Individual
ParentOf
confirms
relates
Name
Married_to


User
Occupation
contains
Individual
ParentOf
confirms
Name
Married_to
id [PK]
spouse_1_id [FK]

## 3.3 Communications Interfaces

The Roots web application will have a network server that is web-based. Therefore the functionality of the application will be created using Javascript (a dynamic programming lanuage designed to be used for the web) and the graphical user interfaces will be initiated through CSS3 and HTML5. The web server exists to retrieve information from the database and provide the user's family tree and information about the individuals on the tree. The product also calls for a database system that stores user information and relationships between users. The HTTP server would be used to store the HTML pages for the application and run the application online. This application should be used on the required web browser(s) (see 2.4 Operating Environment).

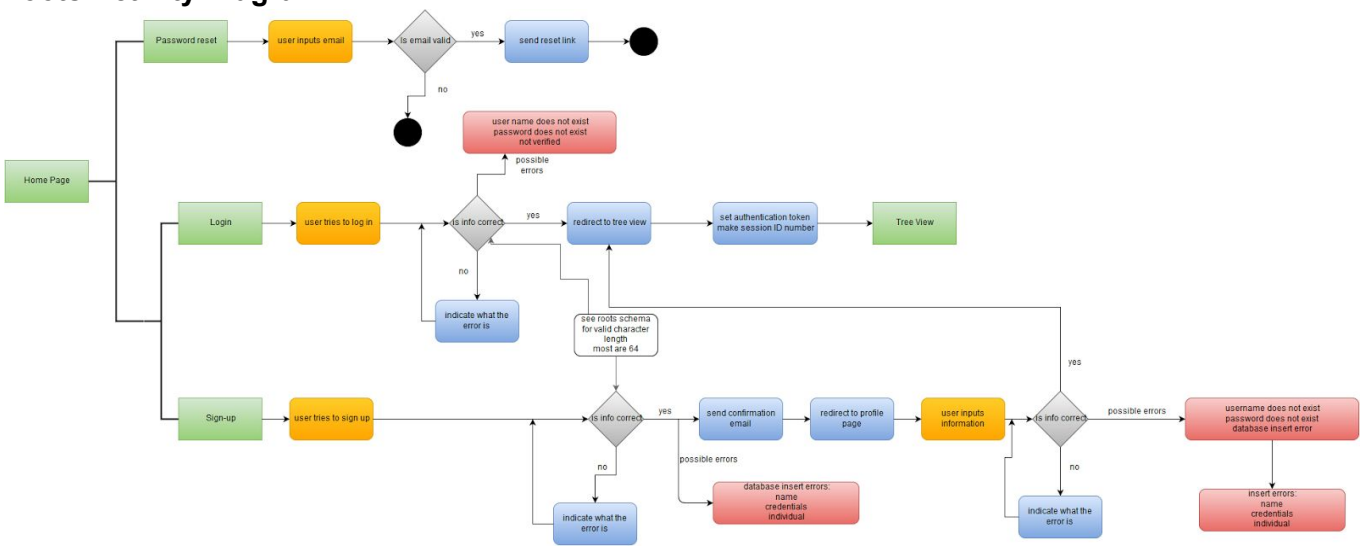
### E-Mail Communication

Users will be prompted to sign-up to use the product through their desired email account. After a breif electronic form is submitted, the user would be sent a link to activate their account. All validated email accounts are accepted for initial sign-up. The user signing up for the product must have access to the valid email account given at sign-up.

### Network Server Communication Protocol

This product will use a series of netwrok server communciation protocols. The application would use node.js modules and angular.js modules to run Javacript. This communication powers the graphical user interface of the application as well as user input validation and user feedback. The application also uses sails; framework that builds scalable, data-driven web APIs with node.js. Sails also provides built-in role-based access control for increased security. It gives both the database and graphical user interface more flexibility due to its high compatiblity.

**Roots Activity Diagram**




# 4. System Features

This section describes the functional requirements for the product through system features and services the product provides.

## 4.1 New user registration

### 4.1.1 Description and Priority

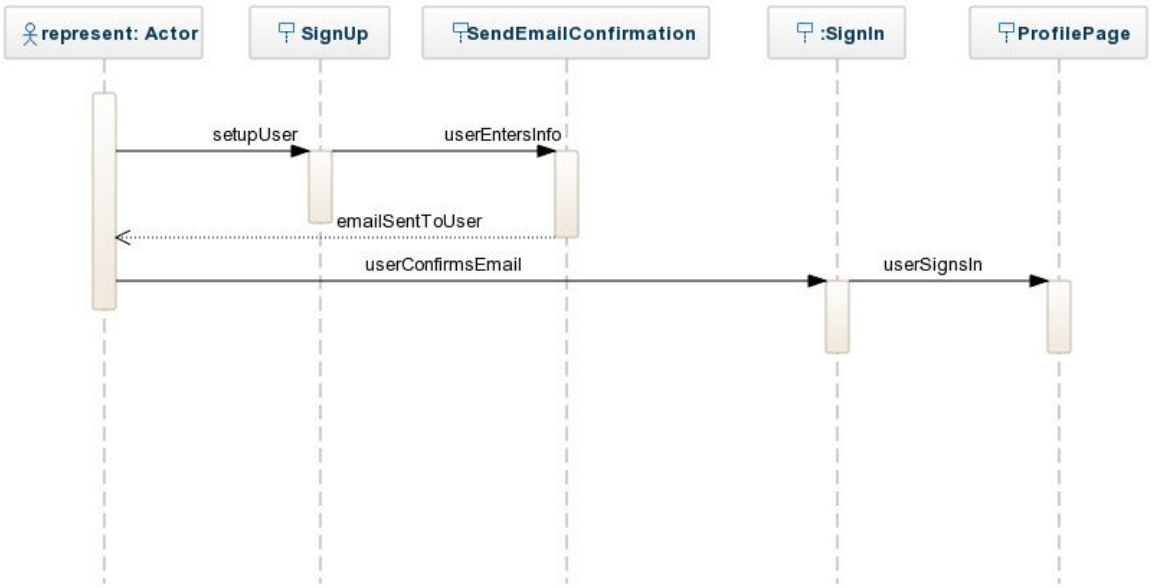
When a user is presented with the login page for the first time, they will input a valid email address as well as a password to sign up for Roots. An email confirmation is then sent to the user. They confirm the email by clicking the link, are signed in, and brough to the “Create Profile” page. It is critical that this be fully funtional since new users must register before they can use our service.

### 4.1.2 Stimulus/Response Sequences

**Step 1:** Application is launched in web browser
**Step 2:** User is prompted to register by entering their first and last name, e-mail address which serves as their user name and creating a password. Users are required to enter password twice to confirm.
**Step 3:** This information is sent to the server and stored in the database.
**Step 4:** E-mail is sent to user’s e-mail account, they are required to click the link within the e-mail to validate their address.
**Step 5:** Once the e-mail address is validated, registration is complete and user is taken to “Create Profile” page.

### 4.1.3 Functional Requirements

REQ-1: All fields are required to be filled out. A “Please fill out this field.’ statement will appear if left empty.
REQ-2: Error response: Password be between 8 and 128 characters.




## 4.2 Returning user login

### 4.2.1 Description and Priority

When the site is visited, users are prompted to either register with the site or log-in if they’re a returning user. This step is critical and must be completed before users can view their tree or add/edit members.

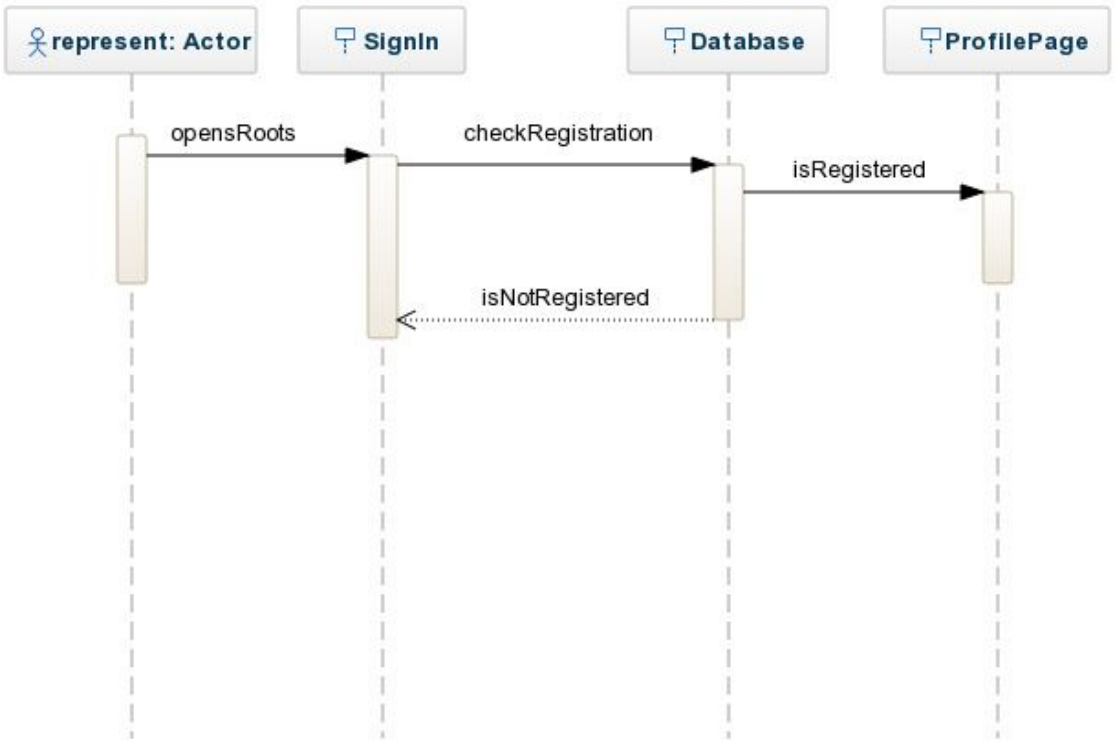
### 4.2.2 Stimulus/Response Sequences

**Step 1:** Application is launched in web browser
**Step 2:** User is prompted to log-in by entering their e-mail address which serves as their user name and password.
**Step 3:** This information is sent to the server and retrieved from the database.
**Step 4:** User log-in is complete.

### 4.2.3 Functional Requirements

REQ-1: All fields are required to be filled out. A “Please fill out this field.’ statement will appear if left empty.

REQ-2: If the user's email address or password is incorrect, the application will display an "incorrect credintials" statement on webpage once the user proceeds to login.




## 4.3 Sign out

### 4.3.1 Description and Priority

This will allow the user to leave their account at any point from any page. After clicking this will direct the user back to the home page, allowing another user to login or sign up. This Feature will be accesable at anypoint during a users session.

### 4.3.2 Stimulus/Response Sequences

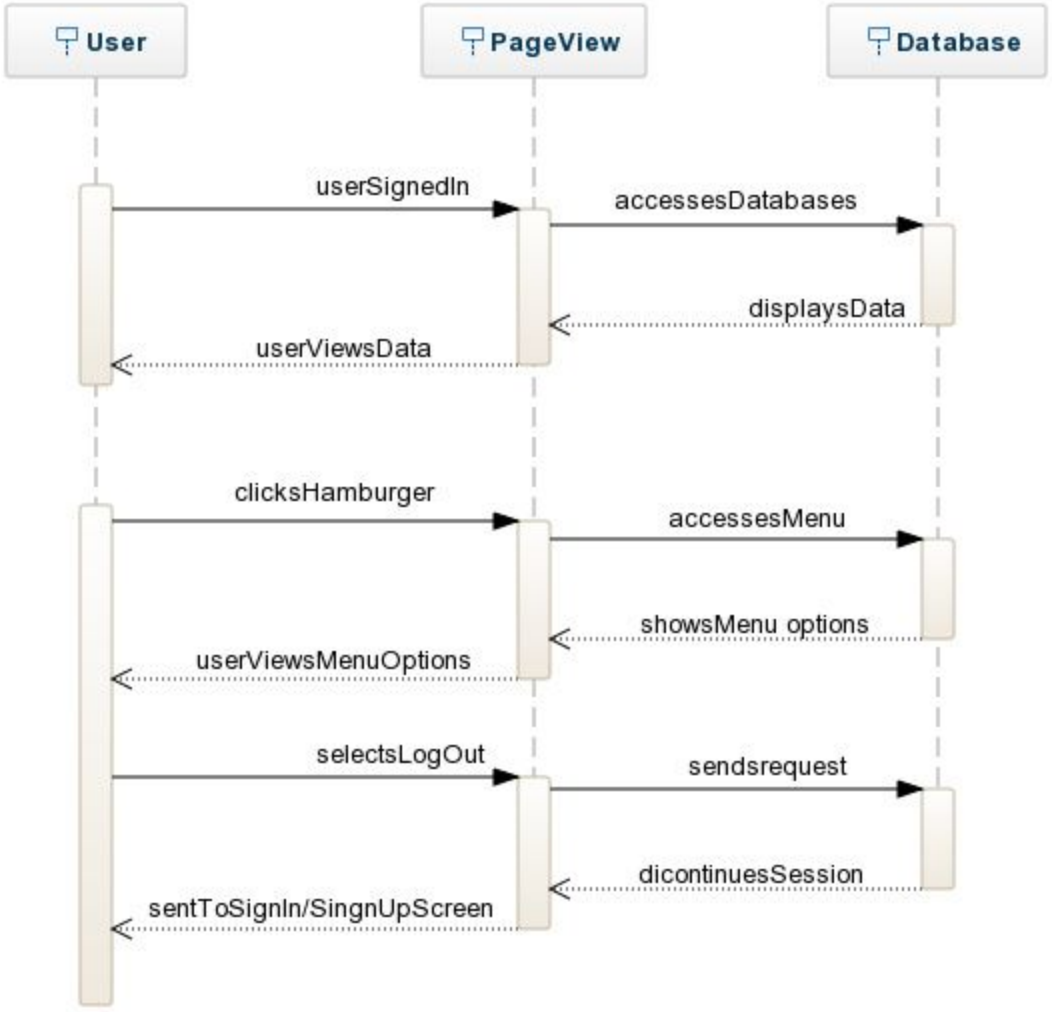
**Step 1:** User clicks on hamburger menu and selects the sign out/ log out option

**Step 2:** Server discontinues session on the user

**Step 3:** User is redirected to home screen to log in / sign up

### 4.3.3 Functional Requirements

REQ-1: Takes user imput of sign out to discontinue session and locks user out of the previous session.




## 4.4 Create profile (new user)

### 4.4.1 Description and Priority

Creation of the user profile is crucial for development of the user's tree. Users complete a bio which allows other users / viewers of that tree can learn more about each individual.

From the Profile screen, users can view their tree or "grow" it by adding family members and creating new branches on the tree which represent these relationships.

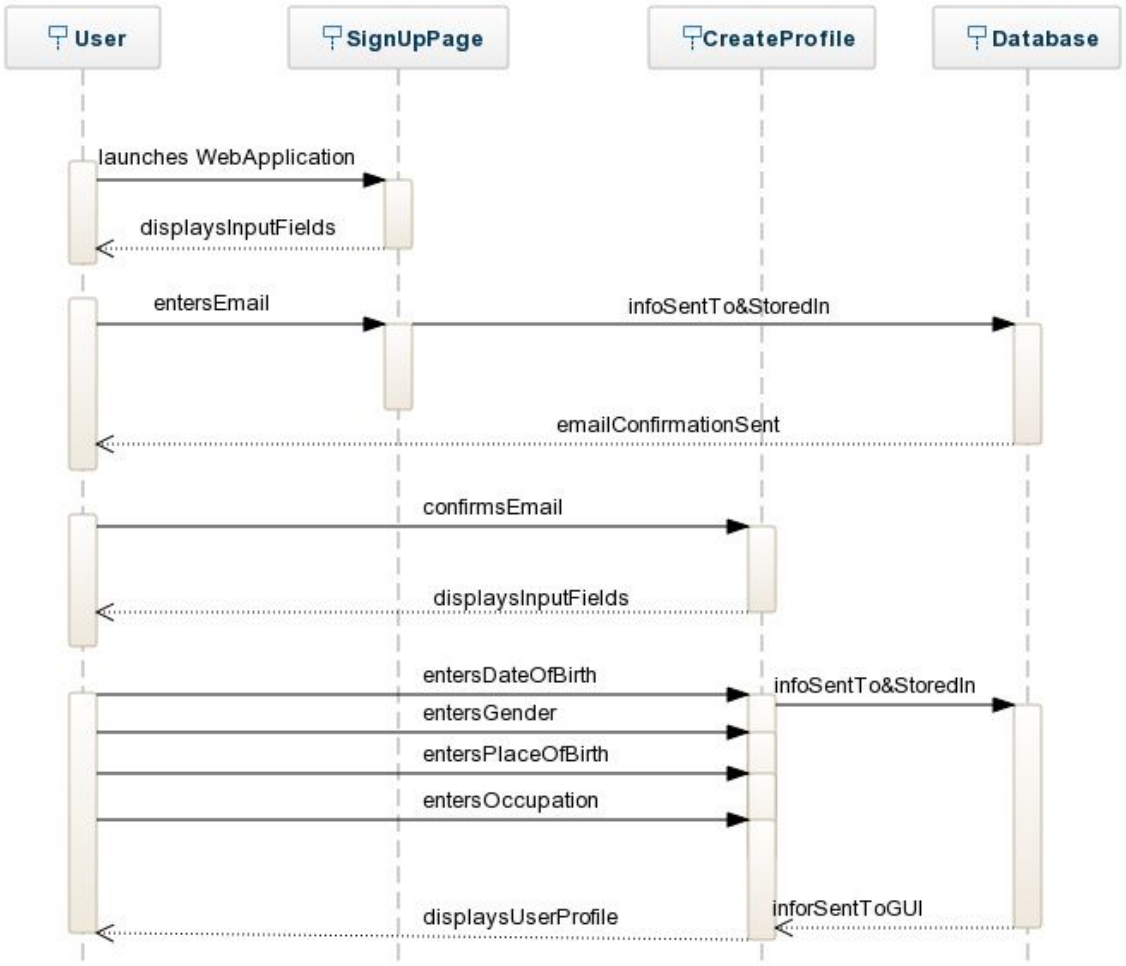
### 4.4.2 Stimulus/Response Sequences

**Step 1:** Application is launched in web browser
**Step 2:** User is prompted to register by entering their e-mail address which serves as their user name and creating a password.
**Step 3:** This information is sent to the server and stored in the database.
**Step 4**: E-mail is sent to user's e-mail account, they are required to click the link within the e-mail to validate their address.
**Step 5:** Once the e-mail address is validated, registration is complete and user is taken to "Create Profile" page.
**Step 6:** "Create Profile" page prompts the user to fill the following fields: Date of birth, Gender, Place of birth (country, state, city), and Occupation. Users can optionally supply a biography and image of themselves

### 4.4.3 Functional Requirements

REQ-1: Data collected in the "Create Profile" page will be stored in a database. The User E-mail, DOB, Gender, Place of Birth, and Occupation and Biography data will be encrypted.
REQ-2: A verification of User E-mail and account registration will be conducted by sending new users an email link to the email used by the user to register the new account that must be responded to within 24 hours time or else link and registration associated with it will expire.

(Sequence diagram below)




## 4.5 Family tree creation

### 4.5.1 Description and Priority

Once the user profile is complete, their family tree is automatically generated, starting with only the user. From here, the user can add family members to their tree. Since this is a family tree application, it is critical that this feature be fully functional.

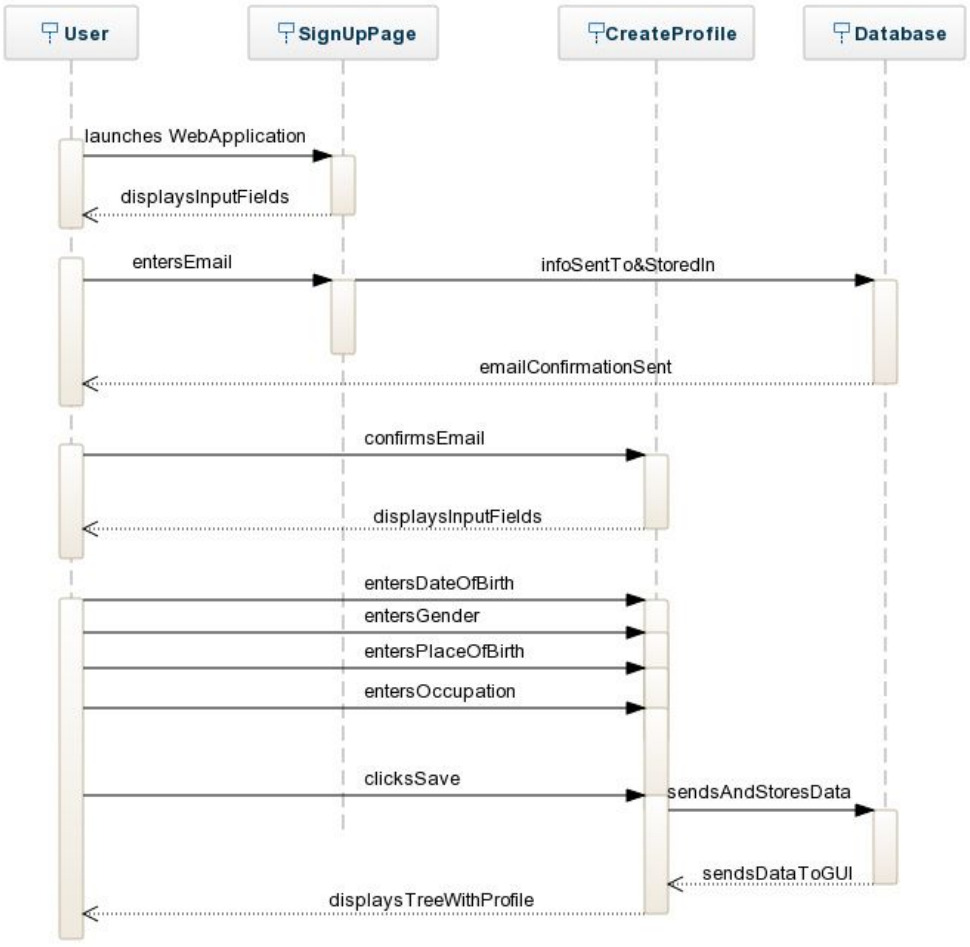
### 4.5.2 Stimulus/Response Sequences

**Step 1:** Application is launched in web browser
**Step 2:** User is prompted to register by entering their e-mail address which serves as their user name and creating a password.
**Step 3:** This information is sent to the server and stored in the database.
**Step 4**: E-mail is sent to user's e-mail account, they are required to navigate to the link within the e-mail to validate their address.

**Step 5:** Once the e-mail address is validated, registration is complete and user asked to log in.
**Step 6:** Upon log in, user is taken to “Create Profile” page.
**Step 7:** “Create Profile” page prompts the user to fill the following fields: Date of birth, Gender, and Place of birth (country, state, city). Users can optionally supply a biography and image of themselves.
**Step 8:** User submits info.
**Step 9:** Info is stored in database and displayed on the left-hand side of the user’s profile page (tree viewer in the middle).

### 4.5.3 Functional Requirements

REQ-1: Takes user information given from user profile creates an object that allows other objects to be created off it in branch fashion
REQ-2: In tree view, users can add, edit and delete family members



## 4.6 Add / edit family member

### 4.6.1 Description and Priority

Once the tree is created, only the user is present, other family members must be manually added. It is essential that this feature is working in order for users to grow their tree.

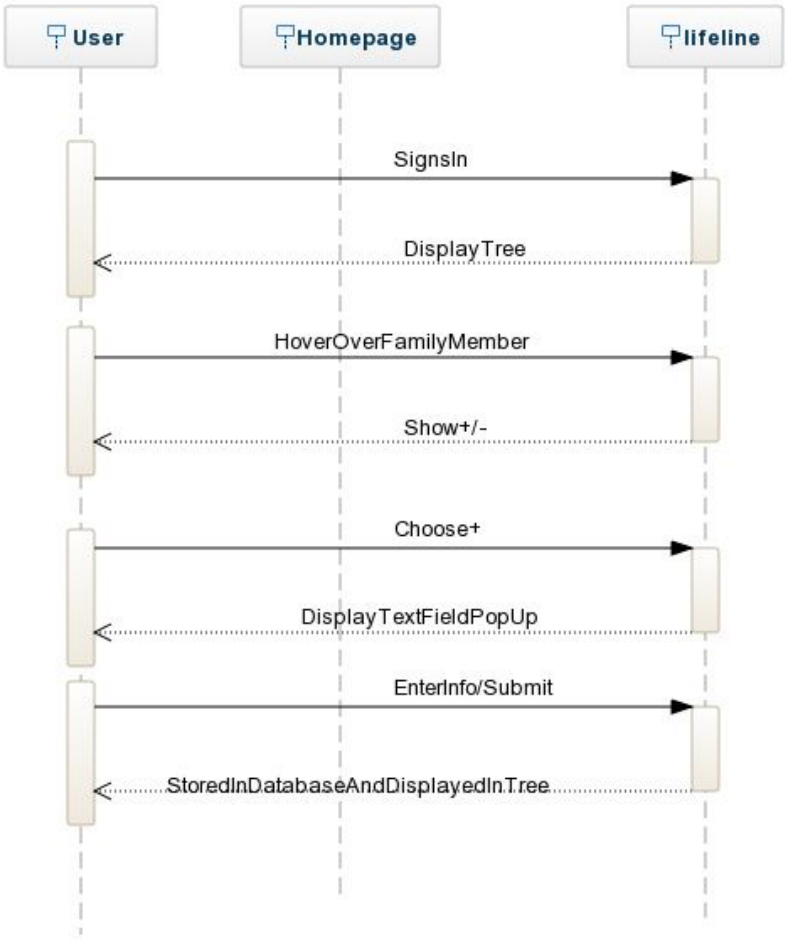
### 4.6.2 Stimulus/Response Sequences

**Step 1:** Application is launched in web browser
**Step 2:** User is prompted to sign into their account
**Step 3:** Once the e-mail address is validated, registration is complete and user is taken to "Create Profile" page
**Step 4:** Display plus or minus sign to indicate "add" or "delete" options
**Step 5:** User selects plus sign / "add"
**Step 6:** User is directed to a new page for "Add Family Member" as a blank profile page with text fields to fill
**Step 7:** User enters family member info into text field and submits (Name, DOB, DOD, POB, Gender, Bio, Photos, Country of origin)
**Step 8:** New family member information saved in database and displayed in tree viewer

### 4.6.3 Functional Requirements

REQ-1: Text field must appear for users to add known information about their relatives. Relationships are not defined by "great" or "grand," they are defined as Mother's Mother (Grandmother) or Mother's Aunt (Great Aunt).

(Sequence diagram below)




## 4.7 Edit or delete family member

### 4.7.1 Description and Priority

Once the tree is created, only the user is present, other family members must be manually added. Users are responsible for “growing” and “pruning” their family tree as necessary, so they are granted permission to edit information for family members on their tree and delete members (ex: accidental additions). This is an important feature but not essential for the software to be fully functional.

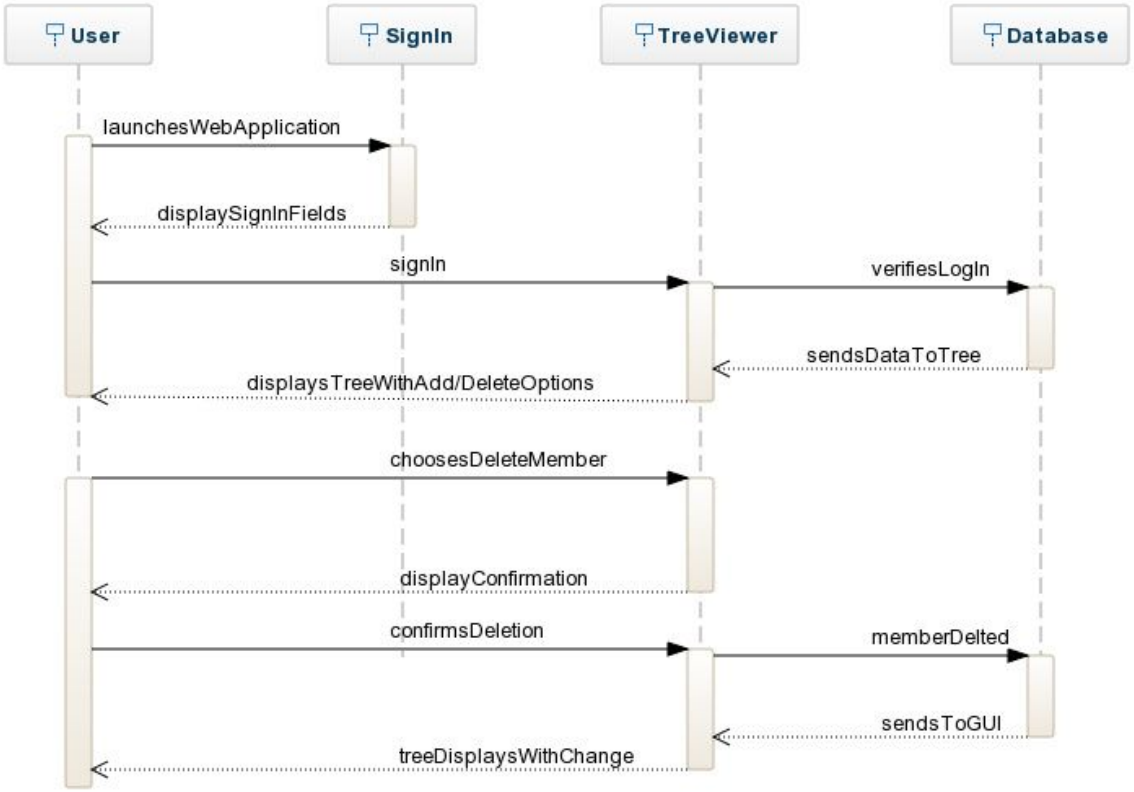
### 4.7.2 Stimulus/Response Sequences

**Step 1:** Application is launched in web browser
**Step 2:** User is prompted to sign into their account

**Step 3:** Tree viewer is automatically displayed upon successful login
**Step 4:** Display plus or minus sign to indicate “add” or “delete” options
**Step 5:** User selects minus sign / “delete”
**Step 6:** Confirm deletion (yes / no)
**Step 8:** Person perged from database

### 4.6.3 Functional Requirements

REQ-2: In side bar, the users press the “trash can” prompting a pop up
REQ-3: Pop up will prompt user to confirm their decission to delete
REQ-4: If prompt up is responded to in the positive the family entry is deleted from database for that family tree.




## 4.8 Display family tree

### 4.8.1 Description and Priority

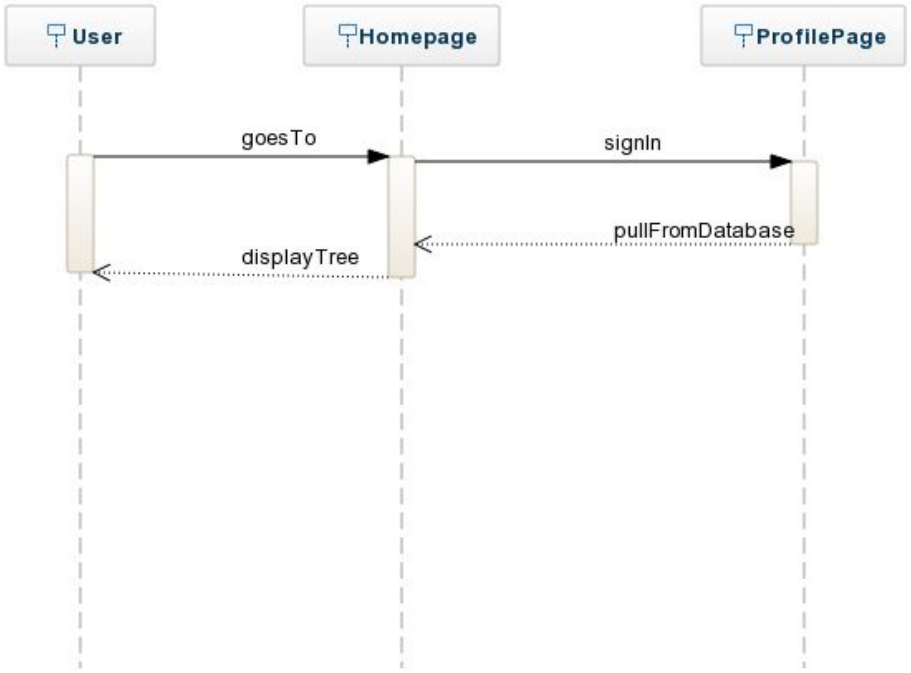
Upon login, users are automatically shown their family tree directly on their profile page. From here, they may view all the information associated with any family member in their tree by clicking on the family member in the tree and viewing their individual bio.

### 4.8.2 Stimulus/Response Sequences

**Step 1:** Application is launched in web browser
**Step 2:** User is prompted to login using their email and password
**Step 3:** This information is sent to the server and retreived from the database.
**Step 4**: Upon log in, user is taken to their profile page which automatically displays their tree.

### 4.8.3 Functional Requirements

REQ 1: At first glance tree should show two generations back based on the user and then pending zoom out should show all the way back to the furthest date in users tree.




## 4.9 Heat Map

### 4.9.1 Description and Priority

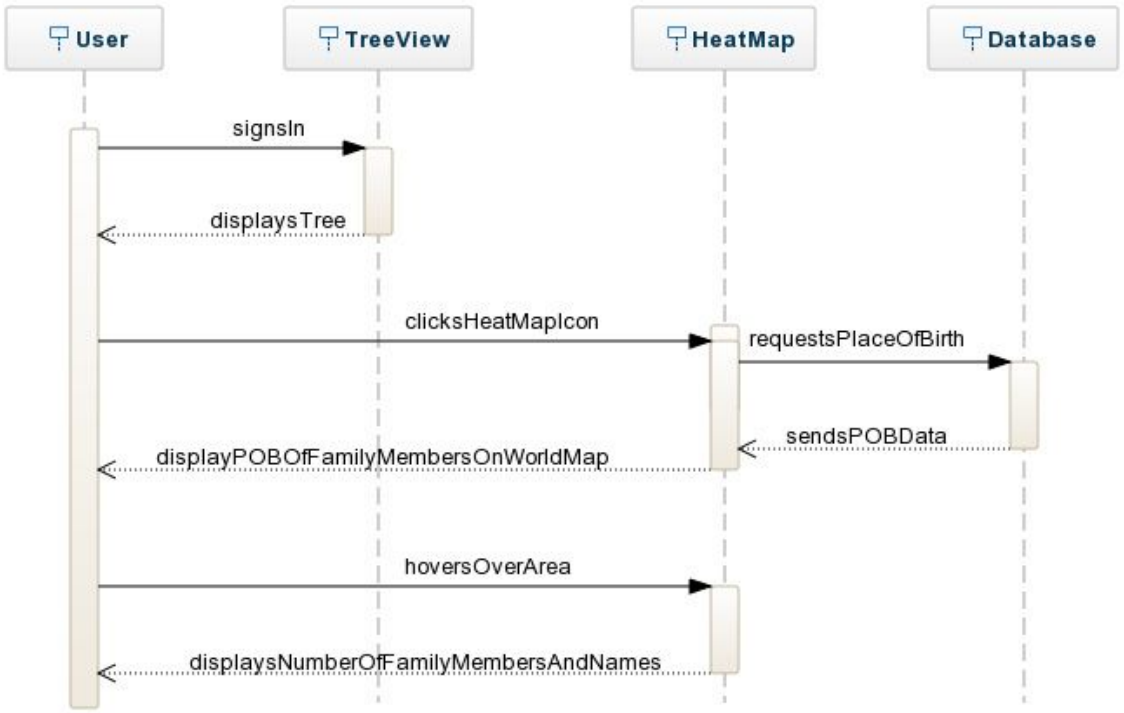
The heat map takes the place of birth of previous generations of the user's family and displays them on a world map. The density of the color is based on how many individuals in the family hail from that region. This is an additional feature, not a crucial element.

### 4.9.2 Stimulus/Response Sequences

**Step 1:** Heat map pulls place of birth from database
**Step 2:** Displays location of Place of birth on world map
**Step 3:** Each consecutive family member from a country or region increases the opacity of color
**Step 4**: User hovers over colored countries
**Step 5:** System displays name as well as number of family members from hovered area

### 4.9.3 Functional Requirements

REQ 1: Takes places of birth of previous generations and displays them on a world map with the shade of color deepening depending on how many members come from that area.




## 4.10 Parliament Chart

### 4.10.1 Description and Priority

The Parliament chart (sunburst chart) is a multi-level pie chart vizualizing hierarchical data of the user's familial relationships. These relationships are depicted by concentric circles.

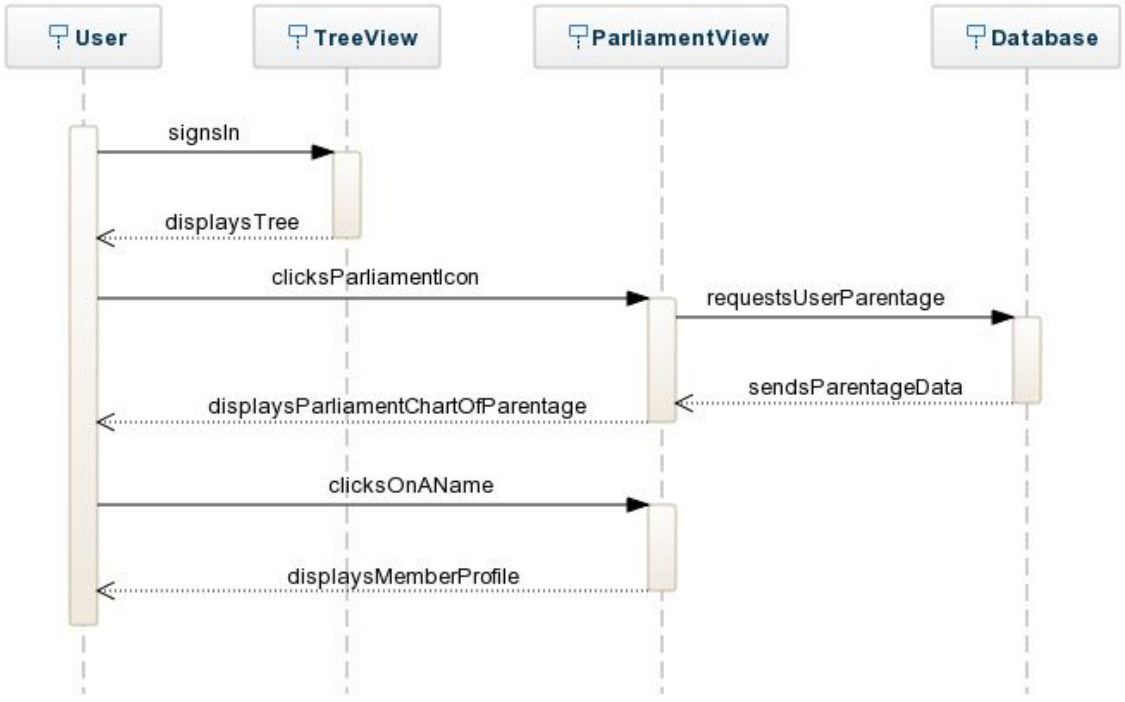
The innermost circle represents the user and from there displays the user's parents and the parent's parents (grandparents), and so forth exponentially. The user can geographically trace the incidence of their family name worldwide.

### 4.10.2 Stimulus/Response Sequences

**Step 1:** Pulls users parents and the parents of each preciding individueal called on from the database
**Step 2:** Creates an object that contains the names of each parent as well each of the previous generations parents
**Step 3**:Displays the names in a circular image( each of which takes half of the previous names circular degree)
**Step 4:** User clicks on a name
**Step 5:** User is brought to that individuals profile (granted they are a member on the user's family tree)

### 4.10.3 Functional Requirements

REQ 1: Parlamentry chart that displays the parents of the the user and the parents of each direct member of their family.



## 4.11 Editing Account Settings

### 4.11.1 Description and Priority

Account users will be able to enable/disable general access to their related information to other account users (name and privacy settings).The privacy settings will control access to viewability of tree nodes, viewability of current account users, as well as access privileges (editing, reading, copying) by outside users to an account user's data.

An Access Control Matrix will be implemented to maintain privacy/user access privileges. Application features that are privacy-control enabled will be listed in the matrix and each access to the application by users will be checked against the matrix for continued access privileges and/ or revocation of privileges of application features.

The option for password changes will be presented at the homescreen of the application. Account user password changes will be made and verified through an email link and verification code sent to the account user's email obtained during the account set up process.

### 4.11.2 Stimulus/Response Sequences

**Step 1:** Stimulus: User selects button at application homescreen to start a change of password
Response: Account Changes/Password Change window is presented
**Step 2:** Stimulus: Application prompts user via popup window for verification of current password
Response: Account user enters current password
Response: User's entry is referenced for validity, if correct proceeds to next step if incorrect reprompts for valid current password entry
**Step 3:** Stimulus: Application prompts user for verification of account user's email address
Response: User's entry is referenced for validity, if correct proceeds to next step if incorrect reprompts for valid account email address
**Step 4:** Stimulus: Account user receives a verification code and a link for secured web connection to application at account email.
Response: User connects via link and signs into account.
**Step 5:** Stimulus:User selects button to start a change of password
**Step 6:** Response: Account Changes/Password Change window is presented option to enter password change verfication code is presented

**Step 7:** Stimulus: User selects entry of verification code and enters code for password change.
Response: User's entry is referenced for validity, if correct proceeds to next step if incorrect reprompts for valid current password entry.
**Step 8:** Stimulus: Application prompts user for new password and presents new password composition guidelines.
Response: User enters new password.

Response: User's new password entry is referenced for validity, if correct proceeds to next step, if incorrect reprompts for valid current password entry.

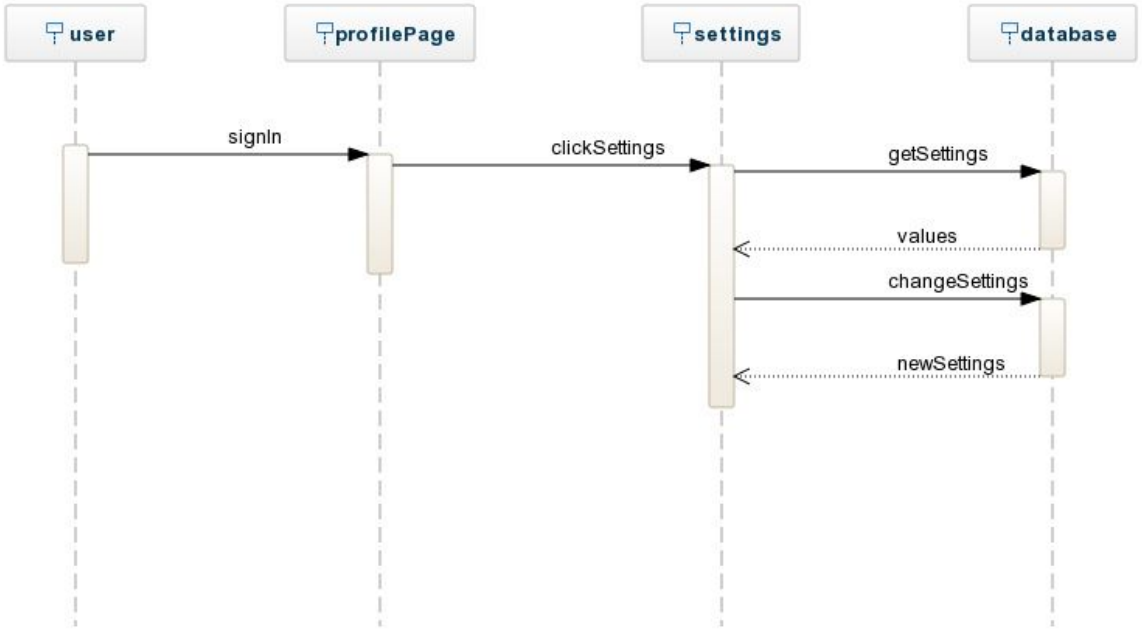
**Step 9:** Stimulus: Application logs user out of current session and presents sign in/homescreen

Response: User logs in with new password

### 4.11.3 Functional Requirements

REQ 1: Users are able to sign up and create a profile

- Requires users name, dob, email, Place of birth, and user profile picture
- Enables to to login anytime they visit site and view tree and profile where they left off
- A family tree that is completely editible by the user, add, delete, modify tree family members




# 5. Other Nonfunctional Requirements

Privacy: There will be a method for users to edit family member's data (users must have permission to edit - registered user) and a method for notifying other users (members on the tree) that changes have been made to member's info on their tree.

We will also provide a method for notifying user 1 when user 2 modifies information for a family member on user 1's tree. If user 2 does not have permission to edit, user 1 will be notified of their attempt.

## 5.1 Performance Requirements

- Whole family tree must load and be viewable by user within 3 seconds of going to the page
- Whole heat map must load and be viewable by user within 3 seconds of going to the page
- Whole parlimentary chart must load and be viewable by user within 3 seconds of going to the page

## 5.2 Security Requirements

**Encryption** of sensitive data will be done using SSL (https) for all aspects involving authentication

**Passwords**
Hashing of passwords with a strong certified, and slow cryptographic one-way hashing scheme. The hashing scheme should be slow since a legitimate user is willing to wait for authentication, whereas a brute force password hack will be more difficult and be slowed considerably by a slower authentication.

Users are encouraged to not minimize the password size (ie: "4 digit pin", but keep it reasonable 35 characters or less)

Password complexity:

- at least one uppercase character
- at least one lowercase character
- one digit
- one special character
- Minimum string length: 8
- Maximum string length:13

Lockout of user account:After three consecutive failed log-in attempts, user will be locked out of the account they're attempting to reach. A link to a requierd password reset is then emailed to the user. (Changing of passwords and account access privileges by account users- see section 4.8.1, pg 26)

## 5.3 Software Quality Attributes

**Quality Requirements**

1. Validate our Requirements Document against customer requirements and ensure all the ISO 25010 requirement groups are adequately addressed by:

   a. Validate that the requirements are consistent; that no two requirements conflict with each other
   b. Check the Requirement Document is correct by confirming that every stakeholder agrees that the Requirement Document adequately reflects all the desired features
   c. Ensure the Requirement Document is complete by confirming that all desired features are included
2. Review all code in review session (or individually), whether in a formal Fagan inspection, scenario-based testing, informal read throughs or some combination
3. As code is generated, ensure that, in a timely manner, it's tested and either passes or is returned to its team with documentation explaining the error(s) found.
4. Ensure that the code is correct by confirming all code passes the following tests:
   a. Method Scope Testing
      i. All-Criterion Test + Data-Flow test + Create Data-Flow Graph
      ii. Boundary Value Analysis & Category Partition Test
      iii. Multiple Condition Coverage
      iv. Boundary Interior Analysis
   b. Class Scope Testing
   c. Flattened Class Scope Testing
   d. Class Interaction Testing
   e. System Testing
   f. User Acceptance Testing
5. Confirm that the code is complete by ensuring the code includes all the features listed in requirements.
6. Interact with teams to promote quality assessment and feedback so that various management decisions can be made causing possible quality improvements that can be implemented.
7. Attempt to measure both efficiency and effectiveness of code being written by reviewing time spent on coding and number of lines of code.
8. Ensure that GUI is accessible for those with disabilities; most notably readable to those with colorblindness by using online tools that display websites as if being seen by someone with colorblindness and, if possible, have someone who is colorblind review the GUI

## Quality Assurance Testing

The QA team is responsible for finding ways to break the program being built in order to refine and improve it. This is done by combing through the code, and testing essential user features to ensure they are working properly.

This is a copy of a QA report for the sign-in function, the following were tests performed:

- Valid login
- Invalid email
- Invalid password
- Username >= 32 characters

- Password >= 32 characters
- Email not verified

| TEST # | Pass/Fail | DESCRIPTION | PARAMATERS | | EXPECTED RESULT | ACTUAL RESULT | | | REASON |
|---|---|---|---|---|---|---|---|---|---|
| | | | email | password | | | | | |
| 1 | Pass | Valid LongIn | rcurran@oswego.edu | notpassword | Tree View | Expected | User is redirected to /treeViewer | N/A | |
| 1a | Pass | Invalid email | abcdefg12345@aol.com | incorrectpass | Stays on login page; "Incorrect credentials." output | Expected | User remains on login page & "Incorrect credenti | N/A | |
| 1b | Pass | Invalid password | rcurran@oswego.edu | incorrectpass | Stays on login page; "Incorrect credentials." output | Expected | User remains on login page & "Incorrect credenti | N/A | |
| 1c | Pass | Username >= 32 | abcdefghijklmnopqrstuvwxyzabc | password | Stays on login page; "Incorrect credentials." output | Expected | User remains on login page & "Incorrect credenti | N/A | |
| 1e | Pass | Password >= 32 | rcurran@oswego.edu | abcdefghijklm | Stays on login page; "Incorrect credentials." output | Expected | User remains on login page & "Incorrect credenti | N/A | |
| 1f | Pass | email not verified | anyone@aol.com | asdfasdf | Stays on login page; "Incorrect credentials." output | Expected | User remains on login page & "Incorrect credenti | N/A | |
| | | | | | | | | | |
| Flow Test | Pass/Fail | DESCRIPTION | PARAMATERS | | PATH | ACTUAL RESULT | | | REASON |
| | | | email | password | | | | | |
| CF1 | Pass | Valid Email | jmaccree@oswego.edu | abcd1234 | User action, router, Login_Controller, anonymus_function_1, user.login, Login_Controller, GUI | Expected | User is redirected to /treeViewer | N/A | |
| CF2 | Pass | Invalid email | jmaccree@store | abcd1234 | user action, router, Login_Controller, anonymus_function_1, user.login, Login_Controller, GUI | Expected | User remains on login page & "Incorrect credenti | N/A | |
| CF3 | Pass | Invalid password | jmaccree@oswego.edu | Ha&&8jfj | user action, router, Login_Controller, anonymus_function_1, user.login, Login_Controller, GUI | Expected | User remains on login page & "Incorrect credenti | N/A | |
| CF4 | Pass | email doesn't exist | ralph@lauren.polo | asdfasdf | user action, router, Login_Controller, anonymus_function_1, user.login, Login_Controller, GUI | Expected | User remains on login page & "Incorrect credenti | N/A | |
| CF5 | Pass | email too long | asdfasfasdfasfasdfasfasdfasfasc | asdfasdf | user action, router, Login_Controller, anonymus_function_1, user.login, Login_Controller, GUI | Expected | User remains on login page & "Incorrect credenti | N/A | |
| CF6 | Pass | password too long | rcurran@oswego.edu | abcdefghijklm | user action, router, Login_Controller, anonymus_function_1, user.login, Login_Controller, GUI | Expected | User remains on login page & "Incorrect credenti | N/A | |
| CF7 | Pass | email not verified | anyone@aol.com | asdfasdf | user action, router, Login_Controller, anonymus_function_1, user.login, Login_Controller, GUI | Expected | User remains on login page & "Incorrect credenti | N/A | |

All tests in this phase passed and the expected outcomes were observed, meaning that all the tests/potential errors were appropriately addressed by the system. For example, if an invalid email or password were to be entered, an error message indicating “Incorrect Credentials” would appear, and did during the test phase.

## Usability Testing

Usability testing was conducted consistently throughout development. The team tested, surveyed and observed potential and real users (with informed consent) on design and functional elements of the product design. User feedback data was collected and used to revise and improve the product. As a result, several iterations of this product have been created. An example of a user test including task lists, surveys and results can be seen below.

### *Scenario*

You are interested in finding out more about your ancestry and have taken it upon yourself to explore the genealogy website myroots.xyz. It’s important to understand this test is examining the website's functionality… not yours! There are no wrong answers!

### *Tasks*

1. You have just found yourself on the site for the first time but are unsure what exactly to expect from the site. Try to find out more information about the site and what it has to offer.
2. After looking around the site signup and make your own account on the site.
3. Now that you have created an account Log in for the first time and try making your profile.

4. After completing your profile you are brought to a new screen try explaining what you think this page is and what you are able to do on it.
5. Try adding new family members to your personal family tree as well as editing your own information or others.
6. Locate your family ancestry heat map, after finding it try and explain what you are currently looking at and what it means to you.
7. Now try the **Parliament map**. Explain what you feel the purpose of this feature is.
8. You are very impressed at this point and want to read the **About** page, go there now.
9. After reading the information on the About page, navigate back to the **Tree Viewer**
10. Log out of your account and instead of logging back in pretend you have forgotten your password, how would you proceed.

*Completion Survey:*

How difficult was it for you to find information about the site? (1 not easy 5 very easy)

1) O 2) O 3) O 4) O 5) O

How did you find the signup process? (1 not easy 5 very easy)

1) O 2) O 3) O 4) O 5) O

If a 1 or a 2 Please explain difficulties:_____

_____

When creating your own profile for the first time did you feel that all information you gave was relevant? (1 not Relevant 5 Very Relevant)

1) O 2) O 3) O 4) O 5) O

If you felt any information was missing from the profile page please list it :_____

_____

How difficult did you find it to add and edit family members to your personal family tree?
(1 not easy 5 very easy)

1) O 2) O 3) O 4) O 5) O

What did you enjoy or find difficult about adding and editing your family tree?:_____

_____

Looking at the heat map what information were you able to obtain from it?:_____

_____

How difficult was it for you to locate and then reset your password? (1 not easy 5 very easy)

1) O 2) O 3) O 4) O 5) O

**Plan B Scenario (Eye Tracker Replacement Plan)**

Assessment of: learnability, efficiency, memorability, affordances

- We can use FiveSecondTest to assess learnability, affordances, and memorability
- We can use Intuition HQ that uses mouse clicks and heat maps to assess these as well, with our incorporated user tasks.

We can also assess user satisfaction in addition to this:

- Sense of enjoyment, fulfillment, and usefulness of site (FeedbackArmy 10 responses for $15)

Accessibility - Cross-browser compatibility, enough HTML usable features, color choice

- W3C markup for checking HTML code, Browswershots (how website looks in different browsers), JuicyStudio

Navigation - should be assessed before site completion, but can still do this

- IA or Information Architecture - how are web pages categorized?
- Findability - enough search boxes, links to finding relevant information? (This would likely fail on our site, so may not be worth testing in first place)
- Number of clicks, how fast - to measure navigation efficiency
- Websort.net, Treejack, WriteMaps to create a sitemap, NavFlow (path analysis, aka how users are moving from one page to another)

It may be possible to do more than one of these, we can tailor our experimental questions and design to better fit the specific topics. Another idea is to have 3-5 users per 4 categories, from HCI department/software design class.

**Results:**

| **Task** | **Proposed Solutions to Errors** |
|---|---|
| **Sign up** | **Urgent:**<br>1. Have password credentials pop up under password fields, once these fields are activated<br>2. Create password credential that is accurate (since 8 characters doesn't work, make sure that it does)<br>3. Make confirmation message more salient, as many users did not notice it and did not proceed to next step right away<br>**Moderate:**<br>4. Create contrast between signup and login buttons<br>5. Simplify password credentials to "password must be 8 to 13 characters and contain letters and numbers"<br>6. Add a message accompanying email confirmation link that says "the link below will redirect you to the login page and confirm your account" to avoid spam, delays, and confusion |
| Log in | None, no issues found |
| **Create a profile** | **Moderate:**<br>1. Remove drop down calendar under |

| | |
|---|---|
| | DOB, or remove year drop down menu portion of calendar<br>**Minor:**<br>2. Refine gender options to “male, female, transgender” |
| **Navigate to heat map and interact with it** | **Urgent:**<br>1. Resolve slow loading time<br>**Moderate:**<br>2. Let user know that map can be dragged: add a prompt message or a tutorial for first-time users<br>3. Let users know the purpose of the map through a tutorial<br>4. When user clicks on continent, this should navigate to a new page with information about the ancestor numbers, in a list<br>5. Remove toggle and reset buttons<br>6. Place zoom buttons closer to map, and make them more salient<br>7. Create a constraint for the drag feature so that the user cannot drag map off the page/make map invisible |
| **Navigate to parliament map and interact with it** | **Urgent:**<br>1. Resolve slow loading time<br>2. Create a key explaining the color code representation - currently the colors hold no meaning and confuse the users (suggestion is to make gender color codes and a key with a bi-color system)<br>3. Have a tutorial explaining the layout and purpose of the map to first-time users, and how it differs from the standard tree view representation<br>4. Rename parliament map “sunburst chart” or “half moon chart” so that it is more clear<br>**Moderate:**<br>5. Allow hover features for this map so that its name comes up<br>6. Have a tutorial explaining the layout and purpose of the map to first-time users, and how it differs from the |

| | |
|---|---|
| | standard tree view representation<br>**Minor:**<br>7. Make sure names do not overlap map borders (aesthetic, visual crowding)<br>8. Add a loading wheel for map loading time |
| **Navigate to the tree viewer** | **Urgent:**<br>1. Make sure the tree loads on the site<br>**Minor:**<br>2. Add a hover feature that has tree viewer name pop up |
| Navigate to the about page | None, expectations met. |
| **Forgot password and how to proceed** | **Urgent:**<br>1. Make message more salient<br>**Moderate:**<br>2. Link in email should have this accompanying message “click on the link to reset your password”<br>**Minor:**<br>3. Rephrase feedback message as “check your email to reset your password” |
| **How to add a family member to your tree** | **Moderate:**<br>1. Add tree and create features on tree page that allows users to add easily, such as extensions on the diagram itself<br>2. Add a “add new family member” button |
| **Other findings leading to needed solutions** | **Urgent:**<br>1. Make sure buttons do not delete profile information<br>2. Make sure green buttons, once other problems are fixed, become functional<br>3. Make sure profile information saves once user enters it (instead of view your tree, have a save button at the bottom of the profile page)<br>4. Make sure pancake menu does not log user out if options are selected<br>5. Make pancake menu “universal” or consistent on every page |

| | |
|---|---|
| | 6. Make sure profile picture loads on homepage<br>7. Make sure pancake menu does not cut off on any pages<br>8. Have toggle features over buttons that allow user to see their functions (“edit your tree” for pencil, “add a new family member” for plus)<br>9. Tree view should be featured on pancake menu<br>**Moderate:**<br>10. Remove trash button and replace with help button<br>11. Remove pancake menu from login and signup pages<br>**Minor:**<br>12. Prompt user for bio section of profile (enter some information about yourself, tell us about your hobbies, nationality, etc) |