

Detailed Requirements (baselined version)

- 1 The web service shall display a home page for the user when the user accesses the service.**
 - 1.1 If the user is not logged in, the web service shall display an Intro page allowing the user to (1) login and (2) create an account.
 - 1.2 If the user is logged in, the web service shall display a home page with the following greeting message: “Welcome!”
- 2 The Intro page shall present an account creation form allowing the user to create an account.**
 - 2.1 The account creation form shall contain the following form fields in the order given:
 - 2.1.1 Name
 - 2.1.2 Email
 - 2.1.3 Username
 - 2.1.4 Password
 - 2.2 These form fields shall be subject to the data requirements in Section 4.1.
 - 2.3 The account creation form shall include a “Sign Up” button below the form fields.
 - 2.4 When the user clicks the “Sign Up” button, the service shall validate the form field values.
 - 2.4.1 The service shall determine whether the form field values are consistent with the data requirements in Section 4.1.
 - 2.4.2 If any form field value is inconsistent with the data requirements, the service shall display a warning message to the user and prevent the user from creating an account.
 - 2.5 If validation is successful, the service shall create an account for the user.
 - 2.5.1 The service shall convert the case-insensitive form field values as specified in Section 4.1 to lowercase.
 - 2.5.2 The service shall insert the form field values in the database as a new ‘user‘ document.
 - 2.6 The service shall prevent duplicate accounts.
 - 2.6.1 The service shall determine if another account exists which has the same username as the username value supplied by the user.
 - 2.6.1.1 If there is a match, the service shall display a warning and prevent the user from creating an account.
 - 2.6.2 The service shall determine if another account exists which has the same email as the email value supplied by the user.
 - 2.6.2.1 If there is a match, the service shall display a warning and prevent the user from creating an account.
- 3 The Intro page shall present a login form allowing the user to login to the service.**
 - 3.1 The login form shall contain the following form fields in the order given:
 - 3.1.1 Username
 - 3.1.2 Password
 - 3.2 The user shall display a button labelled “Login” below the form fields.
 - 3.2.1 When the user clicks the “Login” button, the service shall convert the username value that the user supplied to lowercase.
 - 3.2.2 The service shall then determine whether there exists an account with the lowercase username and password that the user has supplied.
 - 3.2.3 If there does not exist an account, the service shall display an “Invalid credentials” warning to the user and prevent the user from logging in.
 - 3.2.4 If there exists an account, the service shall display the user’s account home page (see Section 4).
- 4 Data Requirements.**
 - 4.1 The service shall enforce the following form field requirements.
 - 4.1.1 Name
 - 4.1.1.1 The name value shall be between 1 character and 30 characters in length, inclusive.
 - 4.1.1.2 The name value shall be case-sensitive.

- 4.1.2 Email
 - 4.1.2.1 The email value shall be unique across all accounts.
 - 4.1.2.2 The email value shall be case-insensitive.
 - 4.1.2.3 The email value shall adhere to the email address requirements as specified by the client-side JavaScript framework.
- 4.1.3 Username
 - 4.1.3.1 The username value shall be unique across all accounts.
 - 4.1.3.2 The username value shall be case-insensitive.
 - 4.1.3.3 The username value shall be between 3 characters and 15 characters in length, inclusive.
- 4.1.4 Password
 - 4.1.4.1 The password value shall be between 6 characters and 12 characters in length, inclusive.
 - 4.1.4.2 The password value shall contain only alphanumeric characters.
 - 4.1.4.3 The password value shall be sensitive.
- 4.2 The service shall store its data in a MongoDB database.
 - 4.2.1 The database shall have a **users** collection to store user data.
 - 4.2.2 The **users** collection shall have fields for storing the respective form field values in Section 2.1.
 - 4.2.3 The database shall have a unique index on each form field specified as unique in Section 4.1.