Software Requirements and Design Document

For

Group 10

Version 1.0

Authors:
Aidan Martin
Riley Corey
Darren Kopacz
Douglas Kendall
James Kerrigan

1. Overview (5 points)

WebSight is a web-application that stores a list of website URLs provided by a user, and sends an alert to that user whenever a website is updated. WebSight has three main web-pages that the user can interact with. When accessing WebSight, there is first a login/create account page so that user data can be saved. Once logged-in, the user can access the main-page to view their currently tracked websites or add new ones. From the main page, the user can log out or access a make-changes page to delete websites or change how often or what specifically they want to track.

When a user adds a website to track, they must provide: a valid URL, preferred notification method (text/email), tracking duration, and specific content to track (words, images, or any change). The main page is designed to be intuitive and quick to navigate, while the make-changes page gives more control and detail to the user.
Alert notifications will include what options the user selected, an option to stop removing notifications for that website, and an option to stop receiving ANY alerts from WebSight for that particular notification method.

2. Functional Requirements (10 points)

LOG-IN PAGE:

1. *HIGH - Ability to create an account or log-in if an account already exists. Else display error messages. [Accounts necessary so that a user's chosen websites can persist indefinitely if they so choose].*
2. *MID - Textbody describing what WebSight is and how it works.*
3. *MID - Enforce password length/character requirements.*
4. *LOW - Forgot password/reset password feature.*
5. *LOW - Bot prevention*

MAIN PAGE:

6. *HIGH - Ability to add a URL, its nickname, and alert option data to a user-specific tracking list (must persist between log-in/log outs).*
7. *HIGH - Log-out button brings user to log-in page.*
8. *HIGH - Check if that URL is valid.*
9. *HIGH - Manage Alerts button links to Manage Alerts Page*
10. *MID - Display X number of URL nicknames in Current Alerts Dropdown (toggles).*
11. *MID - See More button at bottom of Current Alerts. Link to Manage Alerts page.*
12. *MID - Adding a website/Alert instantly updates list or refreshes page to update it.*
13. *LOW - Option to remove a website in the dropdown menu.*
14. *LOW - Each URL listed in dropdown menu links to a sub-menu in Manage Alerts.*
15. *LOW - Ability to switch accounts instead of just logging-out. Dropdown options?*
16. *LOW - Visual Comparisons*

17. *LOW - Ability to display Current Websites dropdown in order of most recently updated, most recently added, etc.*

MAKE CHANGES/MANAGE ALERTS PAGE:

1. *HIGH - Ability to modify any of the options chosen during the add alert menu.*
2. *MID - If no websites added, present option to add a website (links to main page).*
3. *MID - Link back to Main Page.*
4. *LOW - Separate list of "expired" website tracking.*
5. *LOW - Option to re-add an expired website.*
6. *LOW - Change username, password, email, phone.*
7. *LOW - Option to remove ALL saved websites at once.*

3. Non-Functional Requirements (10 points)

Security

1. *Usernames will be unique and require a specific amount of characters.*
2. *All username/passwords will be stored in the database. Eventually, password's will need to be hashed to increase account security.*
3. *User is able to log in with a username and matching password.*
4. *Passwords should display stars as typing.*

Safety

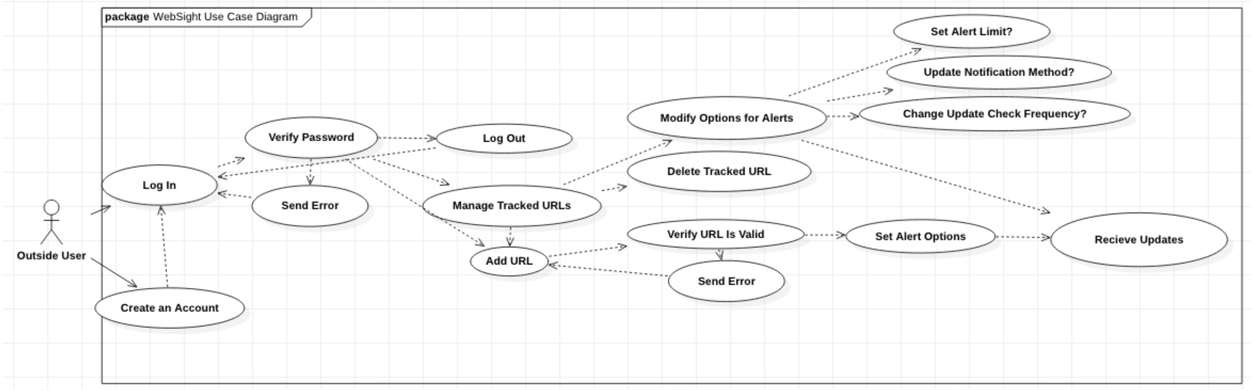
5. *User data involving website URL and tracking options must be saved to each account uniquely ('User' object with urlList object inside).*
6. *Users cannot create an account with an existing username (username list?).*
7. *Users must verify their account by email. This confirms a user will receive email-alerts, and helps prevent misuse of the website update-alert system for spamming or other malicious intent.*
8. *URLs not accessible from the server will return an error*

Performance

9. *Current Websites dropdown should be displayed in order of most recently added.*
10. *Dropdown list implies the website should only display X number of characters before being cut off and followed by '…'. If user chose to give the website a nickname, display that instead, with an internal cap on the # of nickname chars.*
11. *Removing a website from Manage Alerts page must remove it from the user's unique urlList.*
12. *Limit emailing/alerting users when a selected site changes constantly.*
13. *OPTIONAL: Any url added by any user will be saved to a master list. This way, commonly checked websites only need to be checked once per shortest-time interval, rather than having the server check the same website for 100+ times within every minute. However, any time a user deletes any website, the masterList must be checked to see how many people are still tracking it. If less than X users are tracking that site, remove it from masterList and let user's clock-time decide when to check (unless there would be a syncing issue?).*

REQUIRES SYNCING TO SERVER TIME (basically, if user chooses 5 min intervals at 12:03, it will start checking at 12:05).

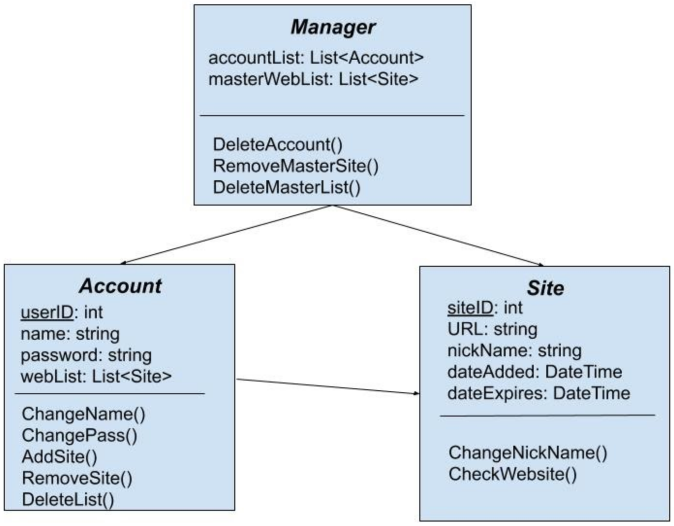
4. Use Case Diagram (10 points)



The homepage offers two options: sign in, or register an account. If the password is verified, the next page presented is for managing tracked URLs. If not, an error is presented on the log-in/home page. Once logged in, the user can add or delete tracked URLs, and modify alert options for previously added URLs.

Adding a URL to the tracked list requires verification of its validity. If valid, the user is presented with alert options, if not, an error is presented.

5. Class Diagram and/or Sequence Diagrams (15 points)



A single manager class instance handles all of the accounts and sites on the application. It interacts with the Account class and Site class to hold lists of Account objects and Site objects. The Account class allows users to create an account that includes their username, email, and password, and is linked to a list of Site objects. This class allows modification of username, password, and site list. The Account class interacts with the Site class, which creates Site objects for each URL added. It includes the website's URL, a nickname, date added, and an expiration date.

6. Operating Environment (5 points)

WebSight is a web-application being developed on the ASP.NET Core framework that will function peacefully on all major browsers, including Safari, Google Chrome, Firefox, and others. We plan to integrate the SendGrid API, use Microsoft SQL as our database for users and their data, both hosted on MS Azure.

7. Assumptions and Dependencies (5 points)

Dependencies currently include the SendGrid API for SMS text alerts and email, and a to-be-determined API for saving website HTML data for analysis.