## **Requirements**

**1. Web app**

**2. Allow selection of any GitHub repository**

**3. Visualize the complete history of the file**

**For each commit**

**4. Contents of the file**

**Metadata for the change, such as:**

**5. Author**

**6. Date**

**7. commit message**

**8 . Size of the file**

**Filter results by:**

**9. File**

**10. Author**

**11. Time frame**

**12. Branch**

**13. File type (language) ?**

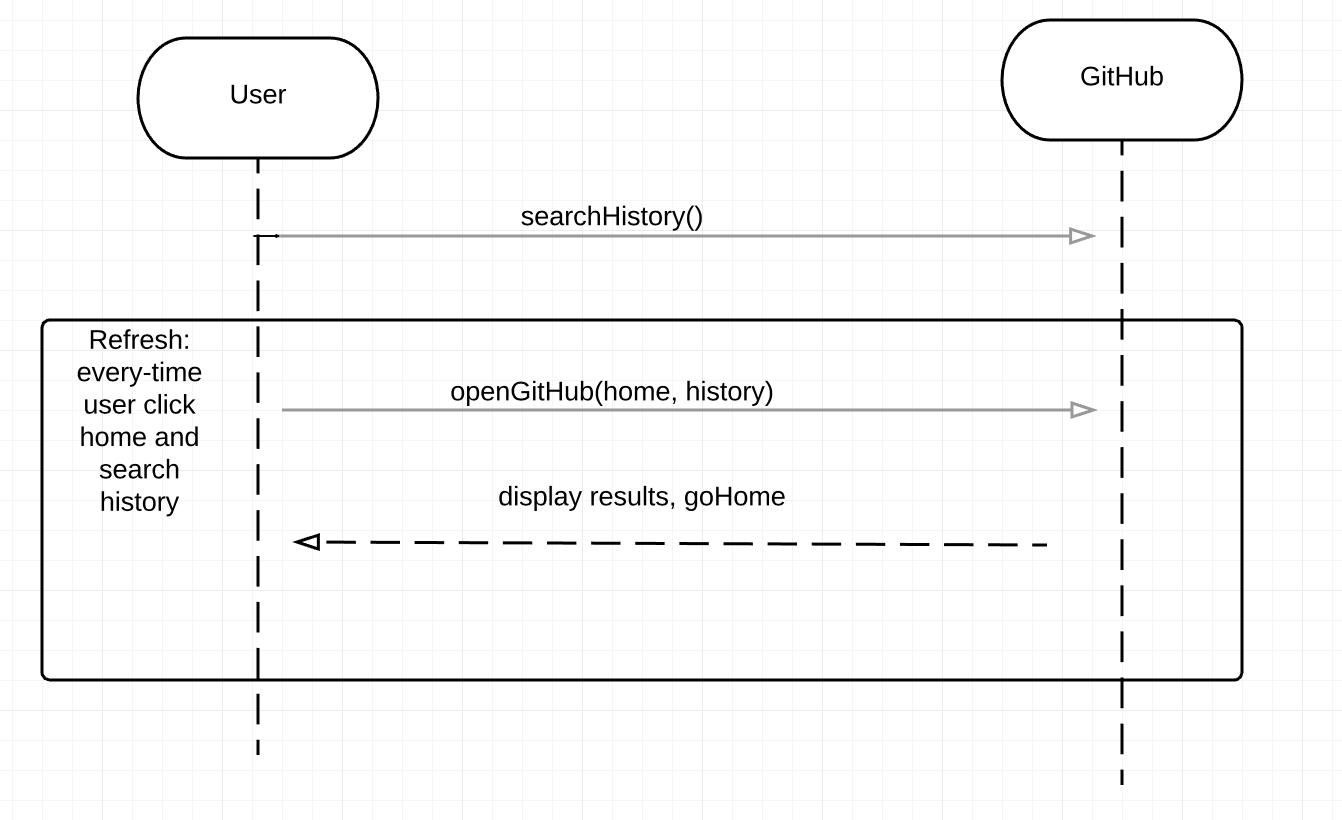
**14. Allow exporting the history of a file to a downloadable file: PNG, CSV, HTML**

**15. Allow login for viewing private repos**

**16. Switch between graphing by commit count and by commit volume**

## 

**Use Case 1: User searches for a public repository (top right)**



**Use Case 2: User gets a list of commits grouped by files, and a stacked graph**

**Use Case 3: User selects one of the other tabs in the top left, grouping by authors or branches**

**Use Case 4: User adds a regex filter for file, author, or branch, eliminating results**

**Use Case 5: User clicks the little “x” in the filter box, removing all filters**

**Use Case 6: User adds a date restriction, eliminating results and confining graph**

**Use Case 7: User selects “volume” or “count” to change the appearance of the F/A/B graph**

**Use Case 8: User selects a file/author/branch from the list (lower left), seeing a commit list**

**Use Case 9: User selects a commit from the list (middle left), seeing detailed commit info**

**Use Case 10: User clicks one of the stacked graph elements, adding an associated filter to the box**

**Use Case 11: Repo state changes while user is viewing something, and the view updates**

**Use Case 12: User views a file that’s been deleted, and the view provides a notification**

**Use Case 13: Exporting the history of a file/author/branch to a file**

**Use Case 14: Exporting the history of a file/author/branch to a file, with filters**

**Use Case 15: Exporting the history of a file/author/branch to a url**

**Use Case 16: Exporting the history of a file/author/branch to a url, with filters**