Team 25: BJDJ

Deliverable #3

Team Repository: https://github.com/CSCD01/team_25-project
Code Repository: https://github.com/CSCD01/focus-android-team25
Kanban board: https://github.com/CSCD01/team_25-project/projects/1

Brennan Law
Johnson Zhu
Jonathan Jarvi
Xu Dong Li

Description of Selected Features

Open New Tab Feature:

The first feature we have selected is the ability to open a new tab more easily. As of now, we believe the only way to open a new tab is to long press on a link that brings up a web context menu from which the option to open a new tab can be selected. Adding the ability to open a new tab in a more straightforward method would help improve usability. One possibility would be to add an “Open New Tab” menu option to the browser overflow menu that opens a new tab when selected. After being selected, the user will be directed to the newly opened tab, but can still access previously used tabs through the tab manager.

Javascript whitelist/blacklist:

The second feature we have selected is the ability to control javascript permissions for individual websites. As of now, we believe the only way to control the javascript permissions for a website is to disable/enable the permissions for all websites. This can become tedious if the user has several websites they regularly use with different javascript permissions. For this feature we could implement a new option in settings, Privacy & Security in the Web Content sections where the block javascript option is currently located. This new option could open a list which would contain the websites where the user wishes for javascript to be disabled. There could also be an option within the page to switch the list from a blacklist to a whitelist. Also the list should override the global javascript permissions.

Feature Designs and Planning

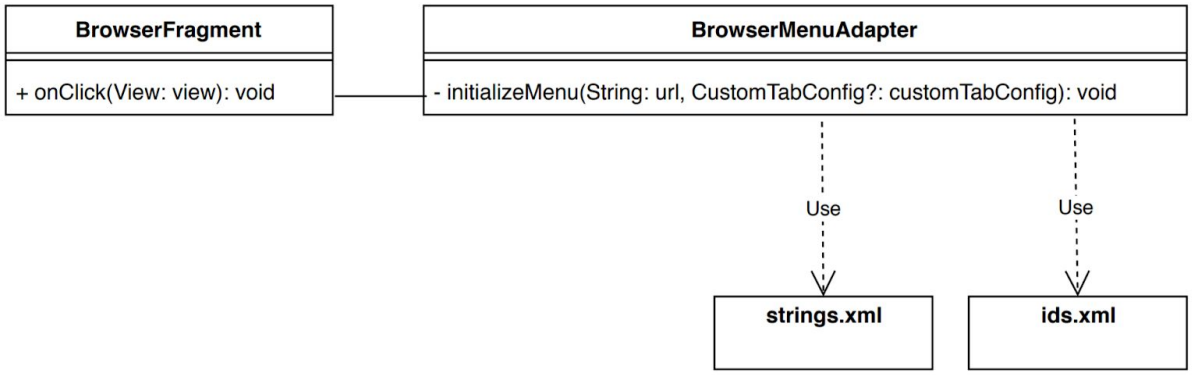
Open New Tab Feature:

The core files that would need to be modified in order to implement the “Open New Tab” feature are as follows:

app/src/main/java/org/mozilla/focus/menu/browser/BrowserMenuAdapter.kt
app/src/main/java/org/mozilla/focus/fragment/BrowserFragment.kt

In order to make additions to the browser’s overflow menu items list, the BrowserMenuAdapter.kt file would need to be modified. This file contains the initializeMenu() function that initializes the browser’s overflow menu when called. This includes creating each of the menu items such as “Add to Home screen” or “Find in Page”. To implement the new feature, we would simply add a new menu item labeled “Open New Tab” along with the creation of the other menu items.

In order to add functionality to this new menu item, the BrowserFragment.kt file would need to be modified. This file contains the OnClick() function that handles on click events for a variety of components found throughout the browser, including the browser's overflow menu and it's menu items. To add functionality for the "Open New Tab" menu item, we would add a new case along with the other on click events, identifying the new menu item by it's view id. Under this new case, we write the functionality of creating a new tab and immediately directing the user to it. Additionally, some components in strings.xml and ids.xml may need to be modified/added in order to accommodate the new UI aspects of opening a new tab.



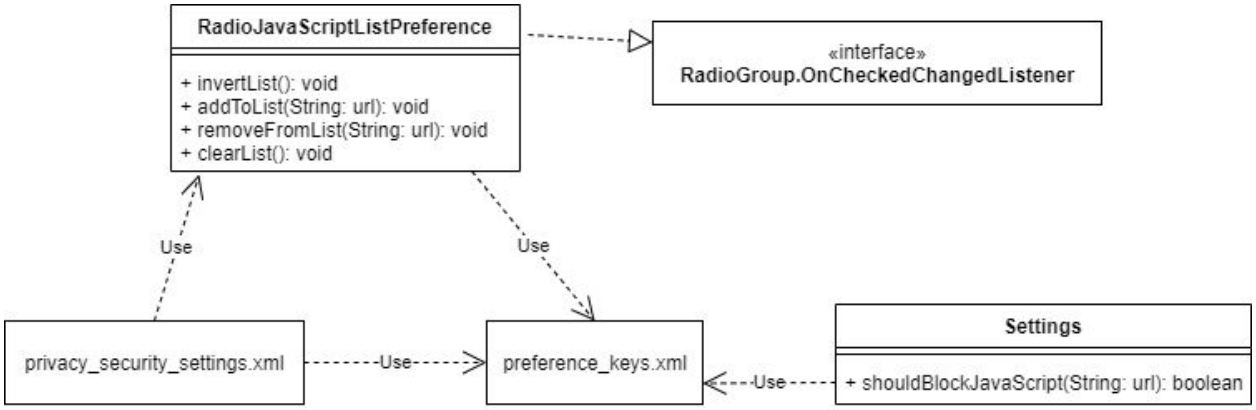
Javascript whitelist/blacklist:

The files that would need to be modified to implement the javascript whitelist/blacklist would be:

app/src/main/res/xml/privacy_security_settings.xml
app/src/main/java/org/mozilla/focus/utils/Settings.kt

In order to make modifications to Privacy & Security, privacy_security_settings.xml must be modified. This file contains the declarations for all the items inside Privacy & Security. To add a new option for a javascript whitelist/blacklist, we would need to add a new tag inside the "Web Content" PreferenceCategory tag. This tag would refer to a custom preference file we would create which would handle the functionality for the option. We would also need to setup a proper user interface to handle the new functions

In Settings.kt we would need to modify the shouldBlockJavaScript method to accept the url that is trying to be accessed to compare it to the list of accepted/blocked websites.



Selected Feature to be Implemented

We decided to implement the Open New Tab feature mainly because it is a major functionality that a browser should have. As a result, it also requires substantial development as we would need to update the UI to include the new menu item as well as update the functionality across several classes to properly support the feature.

Acceptance Tests for Selected Feature to be Implemented

User opens new tab and user is directed to it:

1. Open Firefox Focus
2. Type “beatles” in the URL bar and press enter
3. Open the browser’s overflow menu
4. Press the “Open New Tab” menu Item

Expected Behaviour: A new tab is opened and the user is directed to it, displaying the Focus home screen

User opens new tab then returns to original tab:

1. Open Firefox Focus
2. Type “beatles” in the URL bar and press enter
3. Open the browser’s overflow menu
4. Press the “Open New Tab” menu Item
5. Access list of all existing tabs by pressing the “Tab Manager” icon at the bottom right of the screen
6. Press the first item in the list

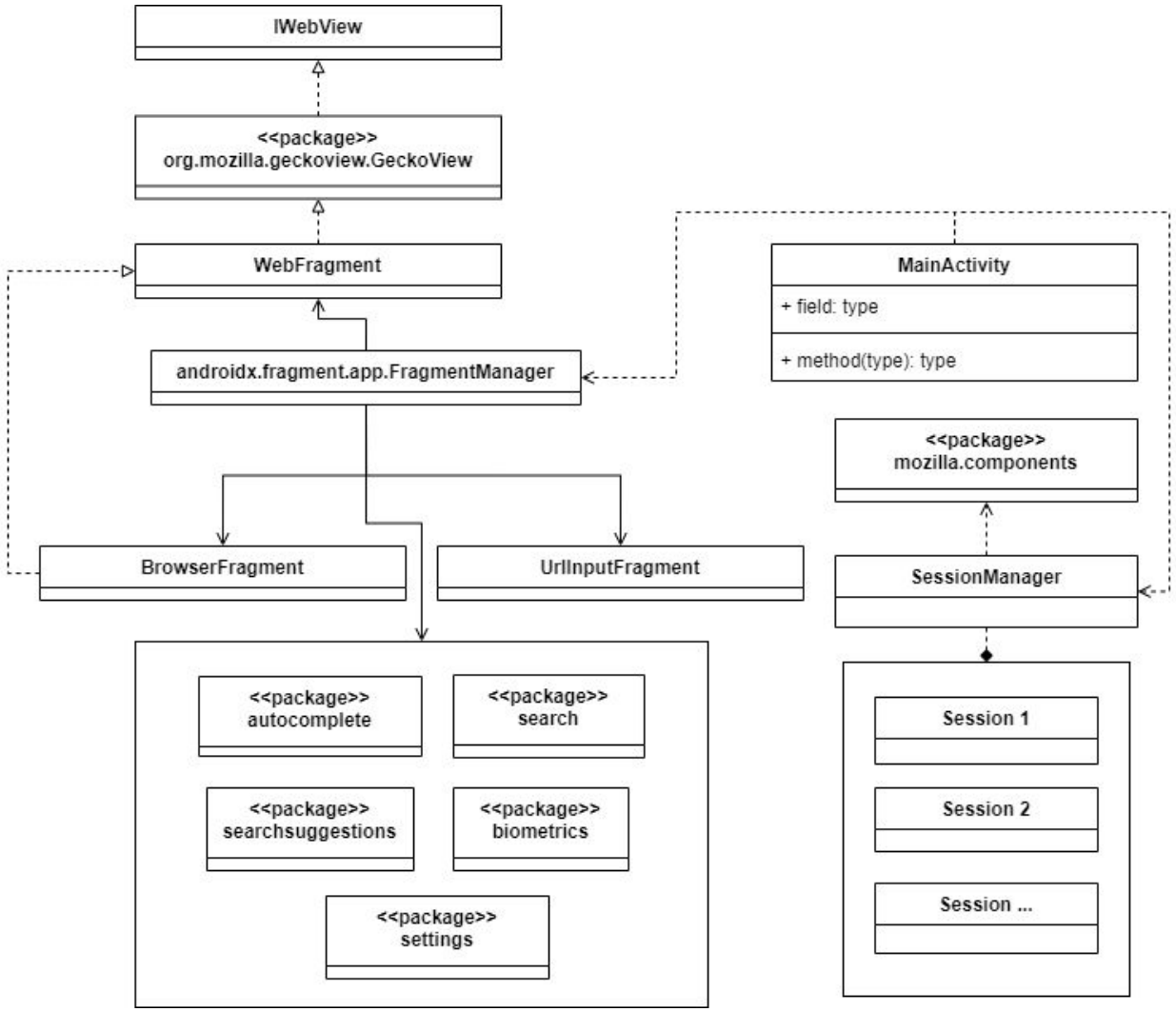
Expected Behaviour: The user is directed to the original tab from which the new tab was opened.

User opens new tab then deletes all browsing history:

1. Open Firefox Focus
2. Type “beatles” in the URL bar and press enter
3. Open the browser’s overflow menu
4. Press the “Open New Tab” menu Item
5. Access list of all existing tabs by pressing the “Tab Manager” icon at the bottom right of the screen
6. Press the “Erase browsing history” item

Expected Behaviour: The user deletes all browsing history including the original tab and the new tab, and is directed to the focus home screen.

High Level Overall Architecture



The MainActivity relies on the SessionManager to create sessions for each tab and uses mozilla.components to handle the processing of the web requests and responses. The FragmentManager manages the different fragment UIs to display the webpage using mozilla's GeckoView and also the other features of the browser such as autocomplete, search, search suggestions, biometrics, and settings.