Deliverable #2

Team 10 - UTSGSC

Kanban Board Link:
https://utsgsc.atlassian.net/secure/RapidBoard.jspa?rapidView=3&projectKey=UFF&selectedIssue=UFF-4&atlOrigin=eyJpIjoiNjY1MGU0ZjM0Y2FkNDJhMjg5YmNmOTgxOTVkZjRlNDUiLCJwIjoiaiJ9

Forked Repo Link: https://github.com/CSCD01-team10/focus-android

List of Potential bugs

1. Focus crashes when tapping the “open and close” button from the settings menu
    - **Link**: https://github.com/mozilla-mobile/focus-android/issues/4342
    - **Estimated effort**: 3 days (1 day for investigation, 1 day for design and implementation, 1 day for testing)
    - **Reproducing issue**:
        a. Open Focus and navigate to any website
        b. Open the settings menu and return to the home screen of your device
        c. select the persistent Focus notification option to “Erase and Open”
    - **Reason for inclusion**: We selected this issue because it was easily reproducible and app crashes are often simple to debug and have the most impact for the stability of the application. This bug specifically has an interesting spin because notifications are generated by a system service in Android which might make it more interesting to debug.

2. Pages no longer render after pushing app to background for an extended period of time
    - **Link**: https://github.com/mozilla-mobile/focus-android/issues/4439
    - **Estimated effort**: 4 days (2 days for research, , 1 day for design and implementation, 1 day for testing)
    - **Reproducing Issue**:
        a. Open a webpage in Focus
        b. Switch to another app by pressing the home button or using "Recent Apps" or any other means
        c. Switch back to Focus
    - **Reason for inclusion**: Interesting bug that is related to application lifecycle and memory usage. It’s a common issue (with several references in the Focus repo) that affects a use case of using the app and minimizing it for a while before reopening it. Unfortunately, it’s difficult to reproduce which makes it difficult to test a solution for it which is ultimately why we didn’t select it

3. Downloading a file with the same name as an existing one, overwrites the previous file
    - **Link**: https://github.com/mozilla-mobile/focus-android/issues/4431

- **Estimated effort:** 3 days (1 day for investigation, 1 day for design and implementation, 1 day for testing)
    - The design and implementation phase may take longer for this bug because of unknown complexity like potentially requiring us to learn how Android applications handle downloading files to the file system.
- **Reproduce Issue:**
    a. Download a file with name filename.png
    b. Observe the date created for the file in a file manager.
    c. Download the file again.
    d. Observe the date created for the file in a file manager.
- **Reason for inclusion:**
    - The reason for including this issue is because there are many users who will download files when they use the web browser and this behaviour could cause someone to unintentionally overwrite a file they previously downloaded with important information.
    - In addition, this behaviour deviates very far from most common day-to-day browsers which will simply append a (x) at the end of the file where x is a number. (For example, filename(1).png)
    - We also want to include this because it lets us challenge ourselves and explore how mobile browsers and Android works with the file system.

4. Bookmarks in autocomplete default to HTTP
    - **Link**: https://github.com/mozilla-mobile/focus-android/issues/4460
    - **Estimate effort:** Unknown - unable to reproduce
    - **Description:** Bookmarks in autocomplete list defaults to HTTP even if the site supports HTTPS.
    - **Reproduce Issue:** We were unable to reproduce this issue
    - **Reason for inclusion:** We selected this issue initially as it seemed important for an application which puts privacy first to default to HTTPS by default hence we were interested in fixing this bug. However, upon further investigation, we realized we couldn't reproduce the issue and opted to investigate another reproducible issue.

5. URL Autocomplete - The "back" action does not redirect the user correctly
    - **Link**: https://github.com/mozilla-mobile/focus-android/issues/4395
    - **Estimated effort**: 2 days (1 day for investigation, 1 day for design, implementation and testing)
    - **Description:** After clicking "Learn More" on URL autocomplete setting page, back button returns user to incorrect page.
    - **Reproducing Issue:**
        a. Open Firefox Focus and navigate to "Settings"/"Search"/"URL Autocomplete".
        b. Tap on the "Learn More" SUMO link.
        c. Tap on the "Back" of the device.

- **Reason for Inclusion:** It is easily reproducible and is not device-specific, so plenty of users would be seeing this issue. Although it is not a pressing issue it is one of those bugs that result in poor user experience.

Chosen Bugs

Issue #1: The app crashes by tapping on 'Open and Erase' button when you are on Settings menu

Issue Link: https://github.com/mozilla-mobile/focus-android/issues/4342

Why we chose this bug

The reason why we chose this issue is because we wanted to work on high impact bug fixes. The high impact comes from the fact that this bug causes Firefox Focus to crash on a user flow that can be quite typical as the user expects consistent behaviour when opening the settings in Focus, exiting the app pressing the "Open and Erase" button from the notifications. Since this involves the system notification bar, this also makes it an interesting bug to fix. In addition, the issue has been opened since August 23, 2019 and seems to have been ignored given the various follow-up comments noting that this is still a problem. The most recent was from January 9, 2020. As a result, we decided to step in to investigate and resolve this long open issue and contribute to restoring the consistent user experience the developers of Firefox Focus aim to deliver.

Estimated effort required to implement change

We divided the process of fixing a bug into 3 separate phases: Spike, Design and Development, and Testing. The Spike is done upfront to investigate and gather information about the bug, following the reproduction steps, examining existing code and determining its root cause. We estimate this to be the most time consuming part because understanding the intents, MainActivity, SettingsActivity and its positions on the back stack will take some time.

Identify risks

Unforeseen complexity when researching and understanding how our app interacts with the system notifications bar since the focus application is only in charge of the notification parameters while the Android system actually constructs the notification. Additionally, this issue can potentially be hard to diagnose because the fatal exception is thrown not by Focus, but by the GeckoView library.

Pull Requests for original and forked repositories

PR for forked repository: https://github.com/CSCD01-team10/focus-android/pull/2
PR for original repository: https://github.com/mozilla-mobile/focus-android/pull/4481

Branch for Fix

The implementation of the bug fix can be found in the branch fix/4342_open_and_erase_crash or by clicking the link below.
https://github.com/CSCD01-team10/focus-android/tree/fix/4342_open_and_erase_crash

Where to find our bug fix implementation

Files changed:

- app/src/main/AndroidManifest.xml

Where to find our bug fix tests:

Files changed:

- app/src/androidTest/java/org/mozilla/focus/activity/EraseAllUserDataTest.java

Acceptance Test Suite

Since this bug isn't that big, our Acceptance Test Suite is not that long.

Test 1: Check all acceptance criteria and expected behaviour is met according to the GitHub issue.
Steps:

1. Clone the repo https://github.com/CSCD01-team10/focus-android and follow the build instructions
2. Open Firefox Focus application
3. Load webpage
4. Tap the 3 vertical dots, Tap Settings
5. Press the home button
6. Pull down the notification tray, expand the Firefox Focus notification and hit "Erase and Open"

Expected Result: You should be taken to the Firefox Focus application with no tabs open, with a message saying "Your browsing history has been cleared."

Technical Commentary

Preliminary research

This includes our findings from the Spike investigations.

The only file changed in this bug fix is **app/src/main/AndroidManifest.xml**

Recall that the components in the MainActivity contains the session manager which is an observable that notifies any observers of changes to sessions.

In the FocusApplication class, we create a NotificationSessionObserver to observe the session manager and handle session manager events

1. Whenever a session is added: The observer starts a new SessionNotificationService which is a service registered with the Android operating system (through the app manifest) and keeps the app alive as long as there is more than a single session. This will show the persistent notification that allows the user to open/erase and open the app.
2. Whenever all sessions are removed: The observer stops the current SessionNotificationService which will effectively "stop" the persistent notification.

For now, we can focus our attention on SessionNotificationService because it handles the action when the user selects the "erase and open" option. If we take a look in the class, we see that the createOpenAndEraseActionIntent method is attached to the option when we build the notification. This method creates a new Android Intent forwards it to the MainActivity with additional data to allow it to call the appropriate methods.

The onNewIntent function in the MainActivity will handle the intent using the processEraseAction function which tells the session manager to remove and close all sessions.

When running the app in debug mode, it looks like the exception is thrown by GeckoView, though I suspect it's related to the lifecycle of the application and how the intent is handled in the onNewIntent function.

Solution research

I have found the root of the problem, which as hypothesized, is related to the lifecycle of the application. To expand, the Android activity class has a lifecycle like many of the other components in the Android framework. When an activity is no longer visible, its stopped.

We have two main activities:

1. MainActivity - the "screen" that displays everything in the browser
2. SettingsActivity - the "screen" used to display the app settings

When we press the settings option from the MainActivity, we'll create a new intent and use it to start the SettingsActivity which will create a new instance of it and push it onto the top of our activity step. Pressing the home button will stop both the MainActivity and SettingsActivity.

Normally, this would be fine, but we configured the launch mode for the MainActivity as singleTop, which essentially tells Android that if we receive an Intent and the activity at the top of the activity stack is configured to handle this, then use it instead of creating a new instance of an activity to handle the intent.

Now, if we take a look at what happens when press the "erase and open" option from the notification, recall that in the earlier exploration, it was found that a new intent was created for the MainActivity to handle. Normally, the MainActivity would be at the top of the activity stack and android will use this to

handle the intent, but the SettingsActivity is now at the top instead! This means a new instance of the MainActivity is created. This causes the “IllegalStateException("Display already acquired")” in the GeckoSession when it tries to acquire the display

Recall the components class contains the session manager and is created by the application class and passed down to its children (ie. activities and fragments) through the context. When a new MainActivity is created, it’ll create an observer for the session manager and try to create a browser fragment for every session. The browser fragments each contain a GeckoView instance used for rendering the page associated with a session.

1. A GeckoView instance is created for rendering a session
2. The GeckoView instance hooks onto its onAttachedToWindow lifecycle event
3. The GeckoView instance will try to acquire the GeckoDisplay (which represents a surface for GeckoView to draw on) for the session
4. Since the session is also associated with a MainActivity instance created earlier, it will already have a session associated with it
5. This will throw the illegal state exception that crashes our app

To fix this, we simply need to update our app so that any activity that can handle the event is used (not necessarily the top) which will ensure our existing MainActivity is always used and fixing the issue.

Acceptance criteria:
Ensure the existing MainActivity is used to handle the new intent for closing sessions and no new instance is created.

Issue #2: URL Autocomplete - The “back” action does not redirect the user correctly.

Issue Link: https://github.com/mozilla-mobile/focus-android/issues/4395

Why we chose this bug

We chose this particular bug as it has a pretty big impact on the settings portion of the app. From initial investigation we found that the “back” action was not just incorrect for the “URL Autocomplete” section after clicking “Learn More” but for all settings “Learn More” button. For a user who is looking at the settings of this app for the first time they would definitely have some questions on how some of these components work, hence clicking the ‘Learn More’ button. After reading this section, the user would want to come back to that particular setting in order to change it if i chose too but because of this bug the user is directed back to the main page.

Estimated effort required to implement change

From reading the issue on github we explored using what the note had said, " It should be the same implementation as for "i" icon for "Add search engine" option." Using this route we had discovered what need to be done in order to fix the bug, and just had to use the same implementation that already existed in the code base for all "Learn More" buttons

Identify risks

Potentially making this change could impact other navigational aspects of the setting activity. We can identify any new issues by running and creating new tests.

Pull Requests for original and forked repositories

PR for forked repository: https://github.com/CSCD01-team10/focus-android/pull/1
PR for original repository: https://github.com/mozilla-mobile/focus-android/pull/4482

Branch for Fix

The implementation of the bug fix can be found in the branch **fix/4395_learn_more_back_redirection** or by clicking the link below.
https://github.com/CSCD01-team10/focus-android/tree/fix/4395_learn_more_back_redirection

Where to find our bug fix implementation

Files changed:

- app/src/main/java/org/mozilla/focus/settings/LearnMoreSwitchPreference.kt

Where to find our bug fix tests

Files changed:

- app/src/androidTest/java/org/mozilla/focus/activity/AccessSettingsTest.java

Acceptance Test Suite

Test 1: Test device backward button after accessing 'Learn More' under 'For Top sites' in Settings/Search/Url Autocomplete

Steps:

1. Clone the repo https://github.com/CSCD01-team10/focus-android and follow the build instructions
2. Open Firefox Focus application
3. Load webpage
4. Tap the 3 vertical dots, Tap Settings
5. Click 'Search'
6. Click 'URL AutoComplete'
1. Click 'Learn More' under 'For Top sites'
2. Click back button on device

Expected Result: Should return back to 'URL Autocomplete' page

Test 2: Test device backward button after accessing 'Learn More' under 'For Sites You Add' in Settings/Search/Url Autocomplete

Steps:

1. Clone the repo https://github.com/CSCD01-team10/focus-android and follow the build instructions
2. Open Firefox Focus application
3. Load webpage
4. Tap the 3 vertical dots, Tap Settings
5. Click 'Search'
6. Click 'URL AutoComplete'
7. Click 'Learn More' under 'For Sites You Add'
8. Click back button on device

Expected Result: Should return back to 'URL Autocomplete' page

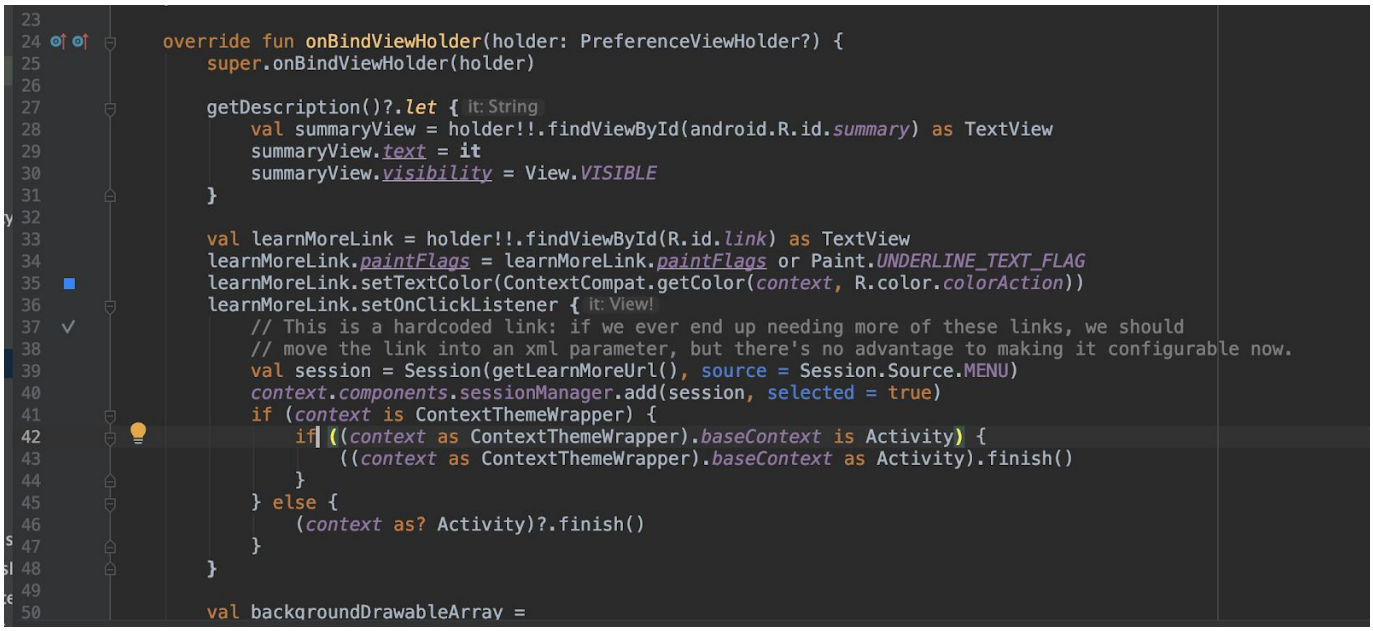
Technical Commentary

To reproduce the bug, click on the Settings -> Search -> URL Autocomplete -> Learn More. When you click the device back button, it goes back to the home page, when it should just take you back to the URL Autocomplete fragment. It seems that this bug can be seen when you click on any "Learn more" link in the app.

The app is split into multiple activities but the ones that we would need to look at are Main Activity and Settings Activity.

- In the Main Activity, there is a function called "OnBackPressed" that is called when the device's back button is pressed
    - Based on which fragment user is currently seeing, the back button will perform differently. This is done in the OnBackPressed button by calling an overridden OnBackPressed function of a specific fragment (SessionSheetFragment, UrlInputFragment, BrowserFragment) based on which fragment is currently viewed.
- When the back button is pressed after clicking on a Learn More link, the "OnBackPressed" function of the BrowserFragment is called.
- BrowserFragment.kt located in /focus-android/app/src/main/java/org/mozilla/focus/fragment/BrowserFragment.kt, seems to be where a fix would be needed to be implemented

When a link is pressed, a new 'Session' is pushed into the SessionManager (as seen on line 39 in the LearnMoreSwitchPrefrence.kt located at /focus-android/app/src/main/java/org/mozilla/focus/settings/LearnMoreSwitchPreference.kt).



```
23
24      override fun onBindViewHolder(holder: PreferenceViewHolder?) {
25          super.onBindViewHolder(holder)
26
27          getDescription()?.let { it: String
28              val summaryView = holder!!.findViewById(android.R.id.summary) as TextView
29              summaryView.text = it
30              summaryView.visibility = View.VISIBLE
31          }
32
33          val learnMoreLink = holder!!.findViewById(R.id.link) as TextView
34          learnMoreLink.paintFlags = learnMoreLink.paintFlags or Paint.UNDERLINE_TEXT_FLAG
35          learnMoreLink.setTextColor(ContextCompat.getColor(context, R.color.colorAction))
36          learnMoreLink.setOnClickListener { it: View!
37              // This is a hardcoded link: if we ever end up needing more of these links, we should
38              // move the link into an xml parameter, but there's no advantage to making it configurable now.
39              val session = Session(getLearnMoreUrl(), source = Session.Source.MENU)
40              context.components.sessionManager.add(session, selected = true)
41              if (context is ContextThemeWrapper) {
42                  if ((context as ContextThemeWrapper).baseContext is Activity) {
43                      ((context as ContextThemeWrapper).baseContext as Activity).finish()
44                  }
45              } else {
46                  (context as? Activity)?.finish()
47              }
48          }
49
50          val backgroundDrawableArray =
```

This step also pushes the MainActivity on top, which is why when the back button is pressed it goes back to MainActivity instead of SettingActvity.

A possible way to solve this would be to create and push a BrowserFragment specifically for the case when the LearnMore button is pressed (handled in the LearnMoreSwithPreference.kt file). This way it is pushed alongside the SettingActivity, and when back is pressed it would return to the SettingActivity. We would also need to ignore the added session in MainActivity so that it doesn't create a BrowserFragment (can be done by using a condition to check what the source of the session is. I.e Source.Menu in this case).

The acceptance criteria of this bug is making sure that, tapping "Learn More" then tapping back should take you back to the settings page from which the user had clicked "Learn More" from (The URL AutoComplete setting screen for this bug). We should also ensure that the back button behaviour everywhere else in the app remains the same as it was before.

Evidence of using our software development process

The following is a screenshot of our Kanban Board. It includes the finished Spike cards for both issues, Development done for and in review and the testing currently in progress. The Spike, Development and Test Cases are subtasks of **UFF-1**, **UFF-5** which are parent issues. These subtasks must be completed in order for an issue to be considered done. Our WIP Limit is 2 cards which is why you can see "In Progress" only has 2 cards. Incremental deployment was done as soon as the Spike, Development and Test Cases subtasks were all in the "Done" column.

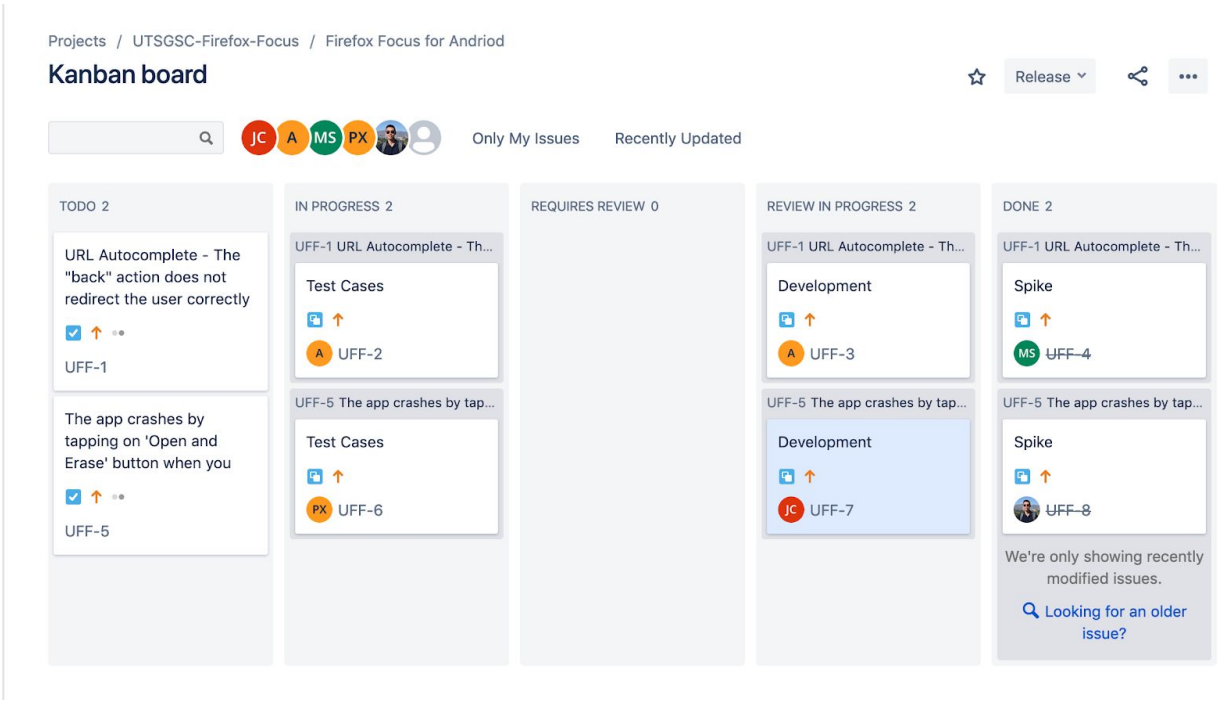


Figure 1: Our Kanban Board while fixing bugs.

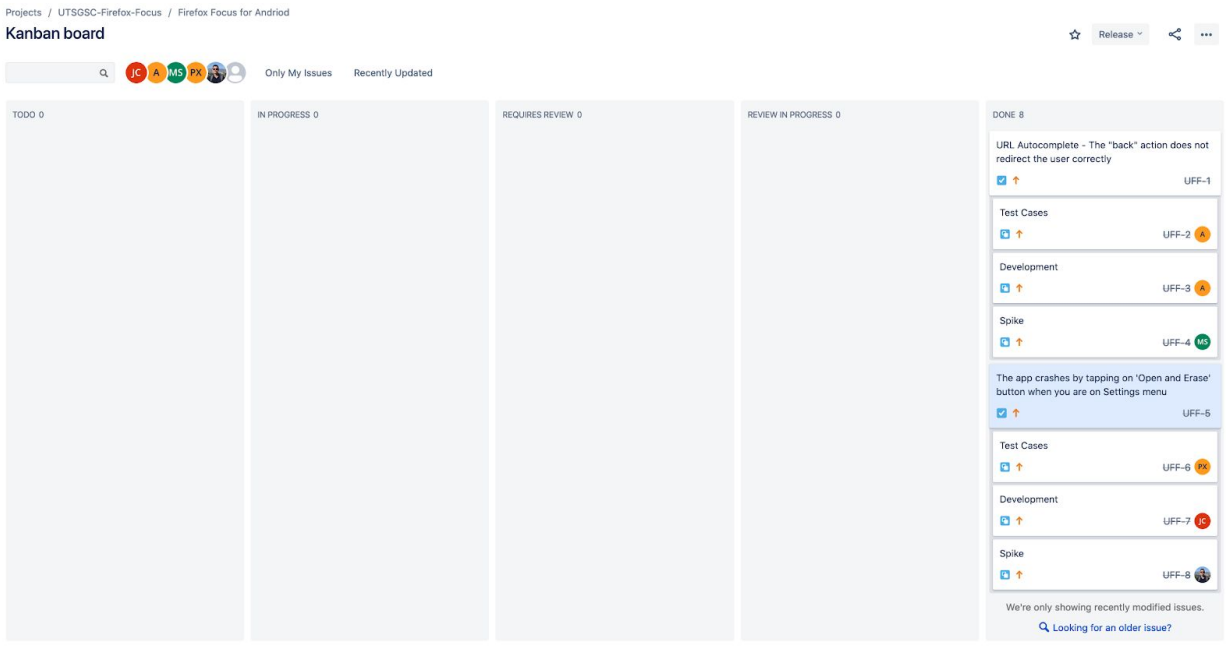


Figure 2: Our Kanban Board after all bugs are fixed.

Meetings and Feedback Loops:

Each of these meetings had every member present. In between these, there were also smaller **daily syncs** in the Facebook Messenger group chat so each the team had visibility into everyone's progress.

- Thursday March 5, 2020: Finalize Choice of bugs and pick 2 bugs to be worked on
- Sunday March 8, 2020: Group Status Meeting
- Monday March 9, 2020: Group Status Meeting

Planned deliverables for each phase

- Specification/Requirements Engineering
    - Document outlining description of the issue, reproduction steps, actual behaviour and expected behaviour.
        - For us, this document is the **GitHub issue page** for the issues we are fixing since they contain this information.
- Design and Development
    - Any Spikes should have documentation to talk about the findings and other details revealed through investigations and spike activities.
        - The results of our spikes can be found in the **Technical Commentary** section of each issue.
    - Development should produce a working bug fix to restore behaviour and a WIP: Pull Request in the forked repo for other members of the team to review.
    - WIP because we need to have tests written and on this branch before proceeding.
- Testing
    - Test(s) that are able to reproduce the original bug and cover any newly introduced code.
    - If development for the issue is approved, testing can be added onto that branch directly. If development for the issue is not approved, a separate branch can be created, reviewed with a PR and merged into the branch using for fixing the issue when approved.