Habitual: A Habit Tracker

Morgan Adrales

University of Missouri – Kansas City

Table of Contents

[Vision Statement 3](#_Toc522714937)

[Requirements 4](#_Toc522714938)

[Product Backlog 5](#_Toc522714939)

[Sprint #1 6](#_Toc522714940)

[Review 6](#_Toc522714941)

Retrospective…………………………………………………………………………………………………………………………………………10

[Sprint #2 7](#_Toc522714943)

[Review 7](#_Toc522714944)

[Retrospective 7](#_Toc522714945)

[Sprint #3 8](#_Toc522714946)

[Review 8](#_Toc522714947)

[Retrospective 8](#_Toc522714948)

[Sprint #4 9](#_Toc522714949)

[Review 9](#_Toc522714950)

[Retrospective 9](#_Toc522714951)

[Sprint #5 10](#_Toc522714952)

[Review 10](#_Toc522714953)

[Retrospective 10](#_Toc522714954)

[Sprint #6 11](#_Toc522714955)

[Review 11](#_Toc522714956)

[Retrospective 11](#_Toc522714957)

# **Vision Statement**

Habitual is an easy-to-use habit tracking app that allows users to set, monitor, and maintain goals. Studies show that goals that are written or otherwise logged are concrete and motivational, and thus more likely to stick. Habitual aims to increase the user’s goal retention by keeping track of these goals and providing feedback on their progress. The key features include:

1. a goal section where goals are created, modified, and if necessary, deleted. The user can also assess/reflect on the progress they’re making through this page.
2. a progress section where Habitual will keep track of daily, weekly, and monthly streaks. The user will also be able to view their goals on a daily, weekly, or monthly basis.

Upon immediate release, Habitual will have the functionality to record goals and accept user feedback on their progress. In future releases, Habitual will be able to track daily streaks on goals and provide encouraging, positive feedback to the user based on their streaks.

# **Requirements**

A discussion of what your application is required to have in functionality. It should identify user roles and goals/actions, and what the key features of the app should be.

|  |  |
| --- | --- |
| **Actor** | **Goal** |
| Customer | Create and set goals on the goal page |
| Customer | Log additional details and progress on the goal page |
| Customer | View goals on a daily, weekly, or monthly basis |
| Customer | Check overall trends in progress and daily/weekly/monthly streaks on the progress page |
| Customer | Receive positive and encouraging feedback from Habitual based on their streaks and progress |
| Customer | Receive timed reports from Habitual regarding their progress (daily/weekly/monthly) |
| Customer | Customize the appearance of the app with preferences such as setting a preferred name, background color, etc. |
| Developer | System shall include a brief, 3-4 step tutorial on how to perform functionality described in stories 1-4 upon the user’s first time opening the app |

### Product Backlog

This will be updated throughout the semester as new PBIs are added, larger items are broken into smaller ones, and completed items removed.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Story ID** | **Story** | **Story Points**  **(in est. hours)** | **Priority** | **Status** |
| 1 | Implement the basic layout of each of the 3 pages: goals, progress, and logging. The user should be able to navigate easily through each of the tabs by swiping | 3 | 1 | Completed in iteration 1 |
| 2 | Add a list view on the goals page | 3 | 2 | Completed Iteration 2 |
| 3 | Add a button to create a goal on the goals page:  On click, the button should open a dialog that prompts the user for more information (i.e. title for goal, duration, description, etc) | 2 | 2 | Completed Iteration 3 |
| 4 | Implement a button on the goal page to remove goals from the list and have them no longer appear on the goal page. User should also be able to select and remove multiple goals at once. | 2 | 1 | Future work |
| 5 | Implement a progress page – user should be able to view their goals on a daily, weekly, and monthly basis. | 3 | 1 | New, future work |
| 6 | Add logging/journaling capability to the goal page – user should be able to edit the goal to add their progress reflection as text that will be displayed as additional goal details. | 2 | 2 | New, future work |
| 7 | Fix issue with Android Studio not running the application due to “installing APKs” running indefinitely | 6 | 1 | Completed iteration 3 |
| 8 | User goals should be added to the list and viewable when adding through the “add new goal” button and dialog | 4 | 1 | Completed iteration 4 |
| 9 | Create a dialog for adding a new goal | 3 | 2 | Completed iteration 4 |
| 10 | Update UI to reflect business changes – goals are now “assignments”, progress is now “calendar”, and logging is now “notes”.1 | 1 | 2 | Completed iteration 4 |
| 11 | Update list to accept an icon, a course name, an assignment title, and a due date as the assignment information (collected from the custom dialog for adding a new assignment) | 4 | 2 | Completed iteration 4 |
| 12 | Add a button to add a reminder on the calendar page | 6 | 3 | Committed iteration 6 |
| 13 | Update UI to match across all 3 pages | 1 | 1 | Completed iteration 5 |
| 14 | Add functionality to add a note on the note page | 2 | 2 | Committed iteration 6 |

# **Sprint #1**

Sprint Backlog

|  |  |  |  |
| --- | --- | --- | --- |
| **Story ID** | **Story / Task** | **Estimated**  **Hours** | **Actual**  **Hours** |
| 1 | Implement the basic layout of each of the 3 pages: goals, progress, and logging. The user should be able to navigate easily through each of the tabs by swiping | 3 | 8 (includes time spent learning java) |
| 1 | Implement a goal page – user should be able to create a goal with the touch of a button and have the goal appear on the page. Goals have the following information:   * Title of goal * Occurrence (daily/weekly/monthly) * Duration (estimated time to complete in minutes) * Additional details (space for user to reflect on their goals or provide motivation for setting the goal, blank by default) | 5 | 0 (broke into smaller tasks, moved to future sprint) |
| 2 | Implement a goal page - user should be able to remove goals from the list and have them no longer appear on the goal page. User should also be able to select and remove multiple goals at once. | 2 | 0 (broke into smaller tasks, moved to future sprint) |

## 

Review



Fig. 1: Screenshot of the Goals page where the user will be able to add/remove goals in a future update.

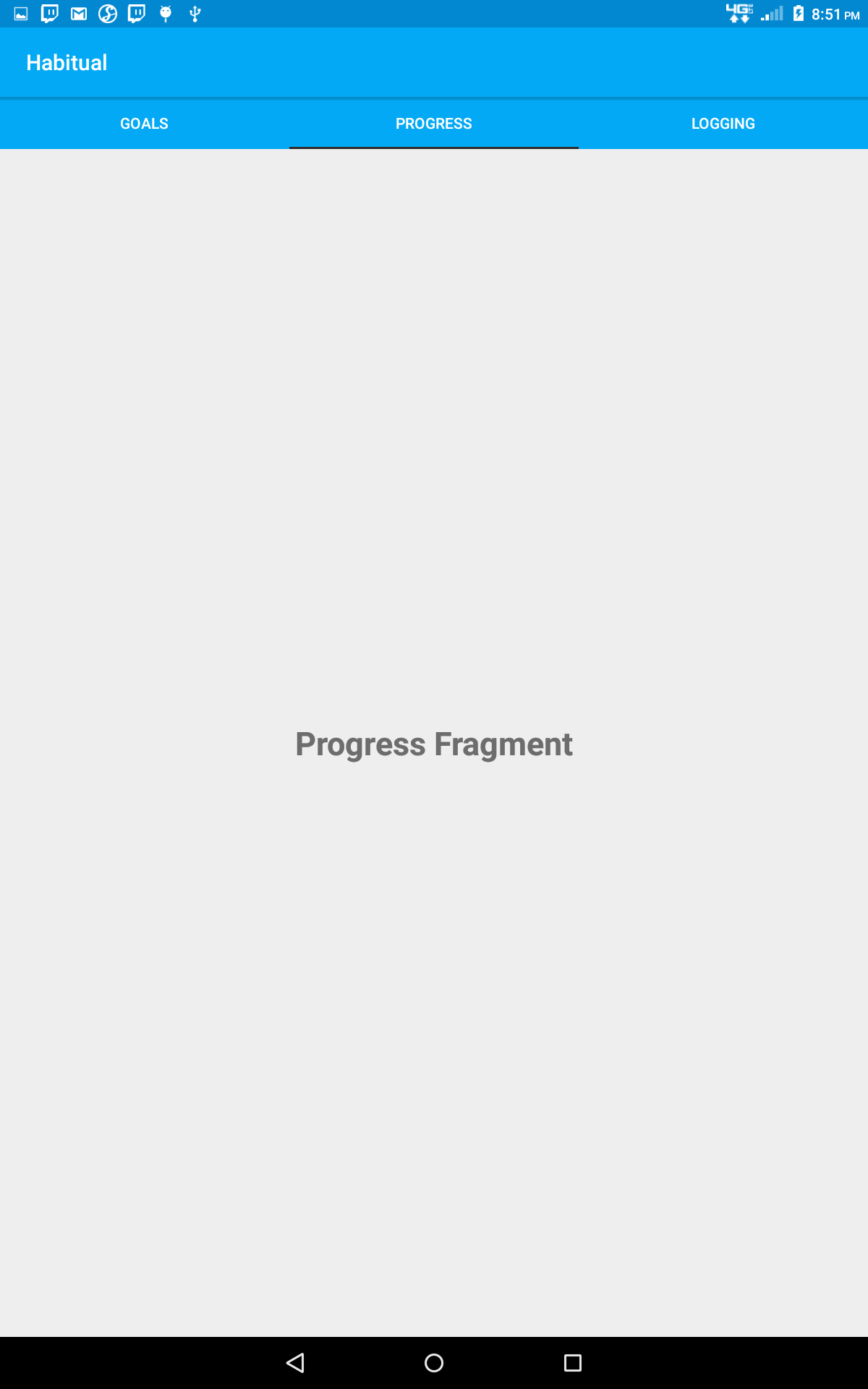


Fig. 2: Screenshot of the Progress page where the user will be able to see their daily/weekly/monthly streaks as well as their targeted or overall progress on the same timeline in a future update.

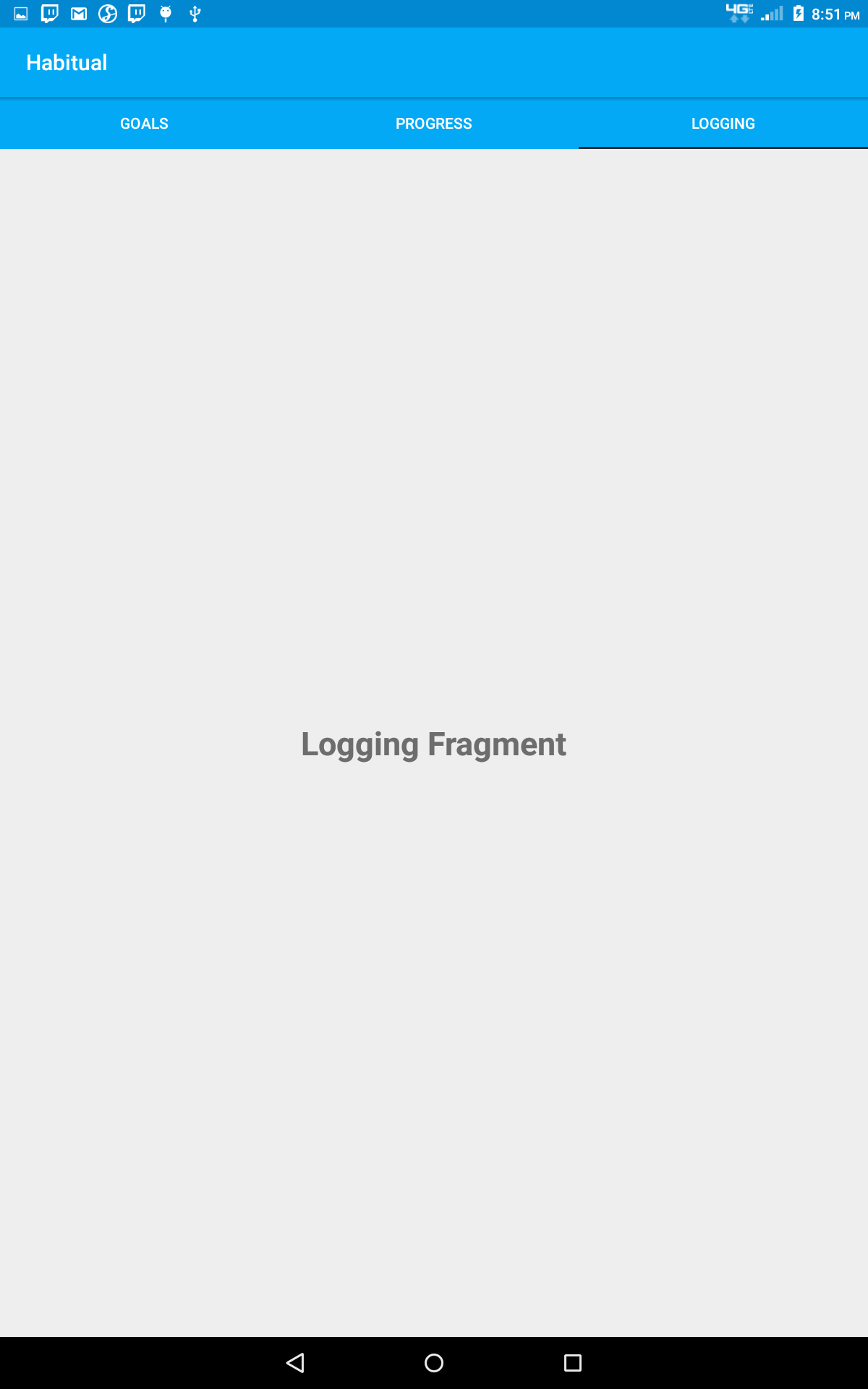


Fig. 3: Screenshot of the Logging page where the user will be able to journal additional details about their goals and progress in a future update.

For this sprint, I initially took on more work than ended up being feasible. Originally, I wanted to implement the Goals page almost entirely and work on the other two in future iterations. I found that that didn’t really make sense. Instead, I needed to setup the basic layout for all 3 pages, then worry about their actual implementation later. So, for this sprint, the work that was actually done included setting up all 3 “pages” (they’re really tabs) and making sure that the user can easily navigate between each one. The user can navigate between each of the tabs pictured above (Fig. 1-3) by swiping left or right.

## Retrospective

There are several ways of implementing a tab-based control but many of the ways that I tried at first weren’t as easy to navigate. For example, I tested a tab layout where the user navigated from page to page by clicking on the tab at the top. I felt that that wasn’t as easy to use and didn’t look as fluid, so I ended up spending 4-5 hours learning how to implement each of these tabs in a way that the user could swipe back and forth between each tab. I struggled with this because learning how to use a tab layout with custom fragments and ViewPager proved to be a cumbersome task to take on. My experience with UI design and app development is limited, so much of this was completely foreign to me. Eventually, I did get the layout that I wanted after a total of 8-10 hours of work. For the next sprint, I plan to replace the placeholder text (i.e. “Goal Fragment”) with some kind of list viewer that will let the user add and remove their goals from the list. I plan to focus on breaking the PBIs into even smaller tasks that will emphasize one feature at a time.

# **Sprint #2**

Sprint Backlog

|  |  |  |  |
| --- | --- | --- | --- |
| **Story ID** | **Story / Task** | **Estimated**  **Hours** | **Actual**  **Hours** |
| 2 | Add a list view on the goals page | 3 | 2 |
| 3 | Add a button to create a goal on the goals page:  On click, the button should open a dialog that prompts the user for more information (i.e. title for goal, duration, description, etc) | 2 | 2 |
| 7 | Fix issue with Android Studio not running the application due to “installing APKs” running indefinitely | 0 | 6 |

## 

## Review

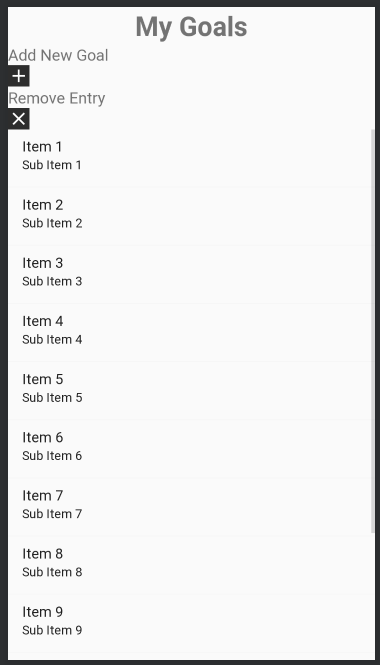


Fig. 4: Goals page as viewed in the Android Studio Designer.

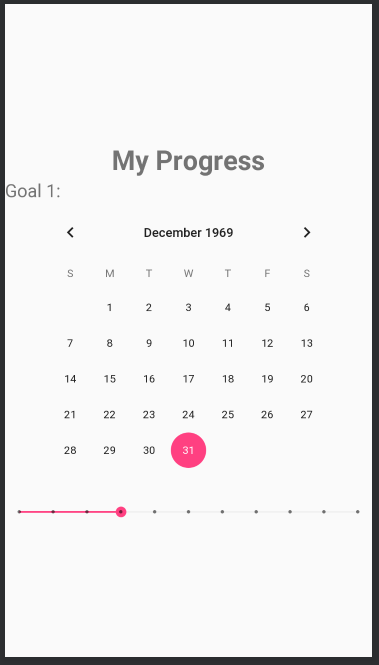


Fig. 5: Progress page as seen from Android Studio designer.



Fig. 6: Journal/Logging page as seen from the Android Studio designer.

For this sprint, I fleshed out more of the basic UI for each of the three pages. From the first sprint, I planned to work on the goals page by adding the ability to add and remove a goal from the list. However, I was unable to complete this goal because Android Studio had an update that prevented me from being able to run my project. In fact, I was not able to run even an empty project. I succeeded in getting my project to build and I know there were not any errors in the code since it ran successfully before the update. However, after updating, uninstalling and reinstalling, and even reverting to an earlier version of Android Studio, I’ve still yet to be able to run a project successfully. I’ve tried several different solutions I found online over the course of 6-7 hours but none have helped so far. I will have to look into it more and hopefully be able to come up with a solution. Instead of implementing the functionality behind the UI, I worked on getting the basic layout and UI features of how I want my application to look. Once I’m able to get my app running again, I will be able to implement the features since I’ll be able to test as I write.

## Retrospective

In this sprint, I ran into a ton of issues with Android Studio that severely stunted my progress. There wasn’t much I could do about it since the update was required and we are also required to use this IDE. I’ve tried several different solutions which included uninstalling, deleting the temp files, and reinstalling as well as turning off instant run and even removing the emulator and setting it up again. Because I wasn’t able to run the application, I wasn’t able to write much of the backend code since I couldn’t test it. However, I think I did a good job of being flexible by making progress in terms of the UI since that can be designed in Android Studio without running the application. In the next sprint, I want my main focus to be resolving this issue while running the application. I’m not sure what I can do to remedy this other than trying it on another machine but that will be my main goal. If that succeeds, then I will return to implementing the backend code for the goals page functionality.

# **Sprint #3**

Sprint Backlog

|  |  |  |  |
| --- | --- | --- | --- |
| **Story ID** | **Story / Task** | **Estimated**  **Hours** | **Actual**  **Hours** |
| 7 | Fix issue with Android Studio not running the application due to “installing APKs” running indefinitely | 6 | 8 |
| 2 | Add a list view on the goals page | 3 | 2 |
| 3 | Add a button to create a goal on the goals page:  On click, the button should open a dialog that prompts the user for more information (i.e. title for goal, duration, description, etc) | 2 | 5 |

## Review

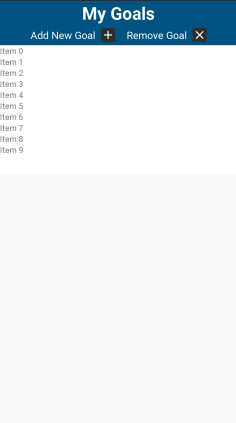


Fig. 7: Updated Goals Fragment page.

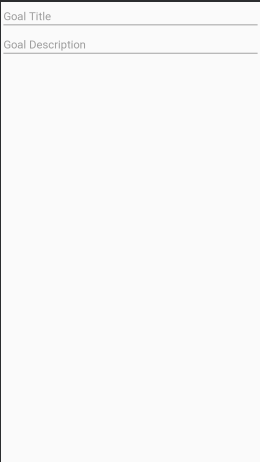


Fig. 8: Custom input dialog for adding a new goal.

In this sprint, I added some of the backend functionality for adding a new goal to the goals fragment. Now the user has the ability to press the “Add new goal” button featured in Figure 7 that will bring up a custom dialog. This dialog prompts the user to enter additional information relevant to the goal including the goal title and description. This information then gets stored in the list view of all of the future goals. I planned to be able to add additional information to this dialog such as the duration or frequency that the goal appears, but I was not able to get to that implementation yet because those items require either choosing from a list of a date picker. Storing that data is a bit trickier than just storing strings, so I will need to do more research into how to convert/store that data into a list.

## Retrospective

I spent a good amount of time this sprint working on fixing my Android Studio. As I mentioned in the previous sprint, I was not able to complete much because Android Studio updated and no longer could build even an empty project. After several hours of attempts to get Android Studio to compile on my previous machine that included a total uninstall, reinstall, and deleting of the cache, I was forced to obtain a different machine to develop on Android Studio. For whatever reason, the fresh install on a new machine with different hardware has solved the issue, although getting a new machine was a hassle and not an efficient solution.

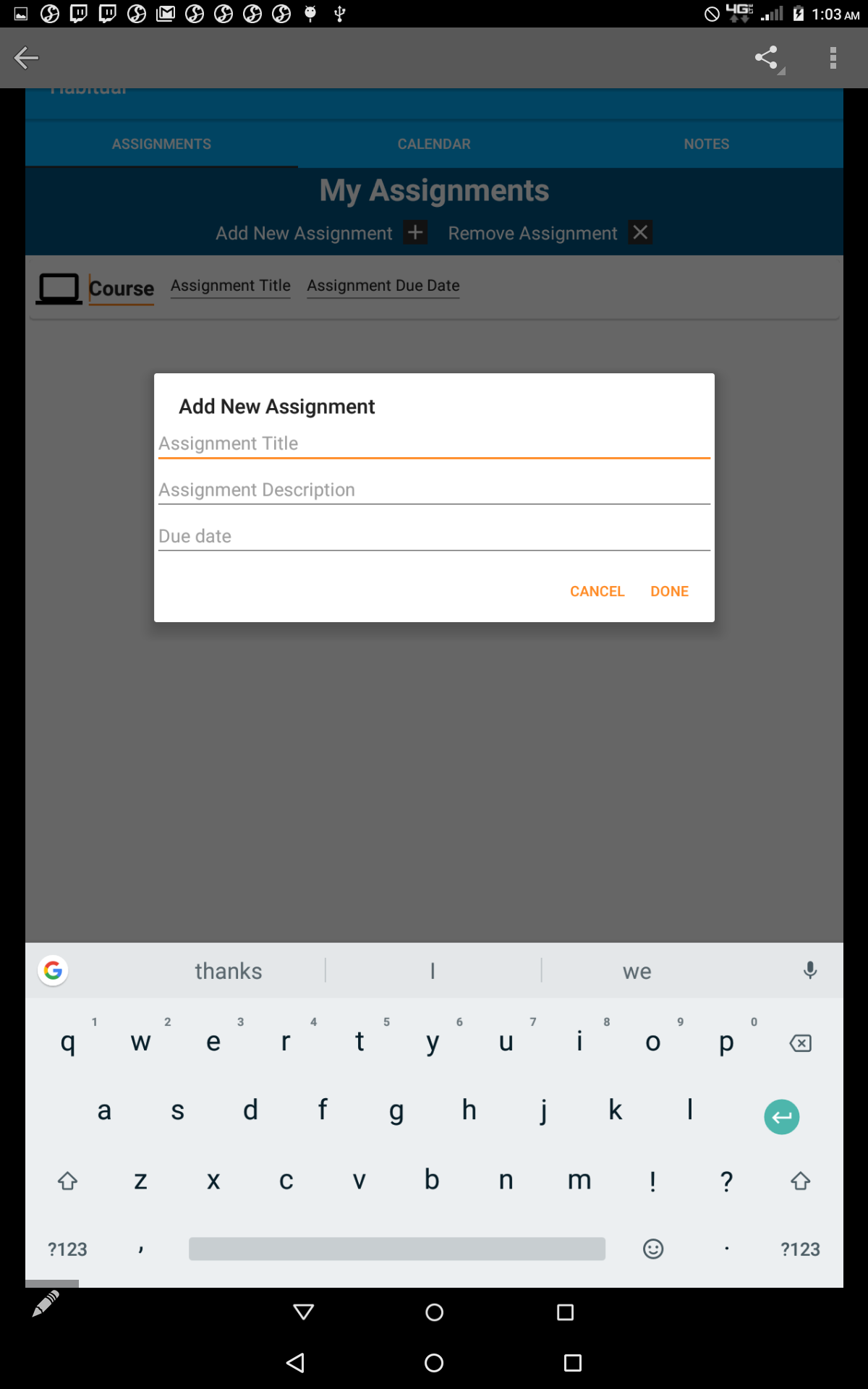
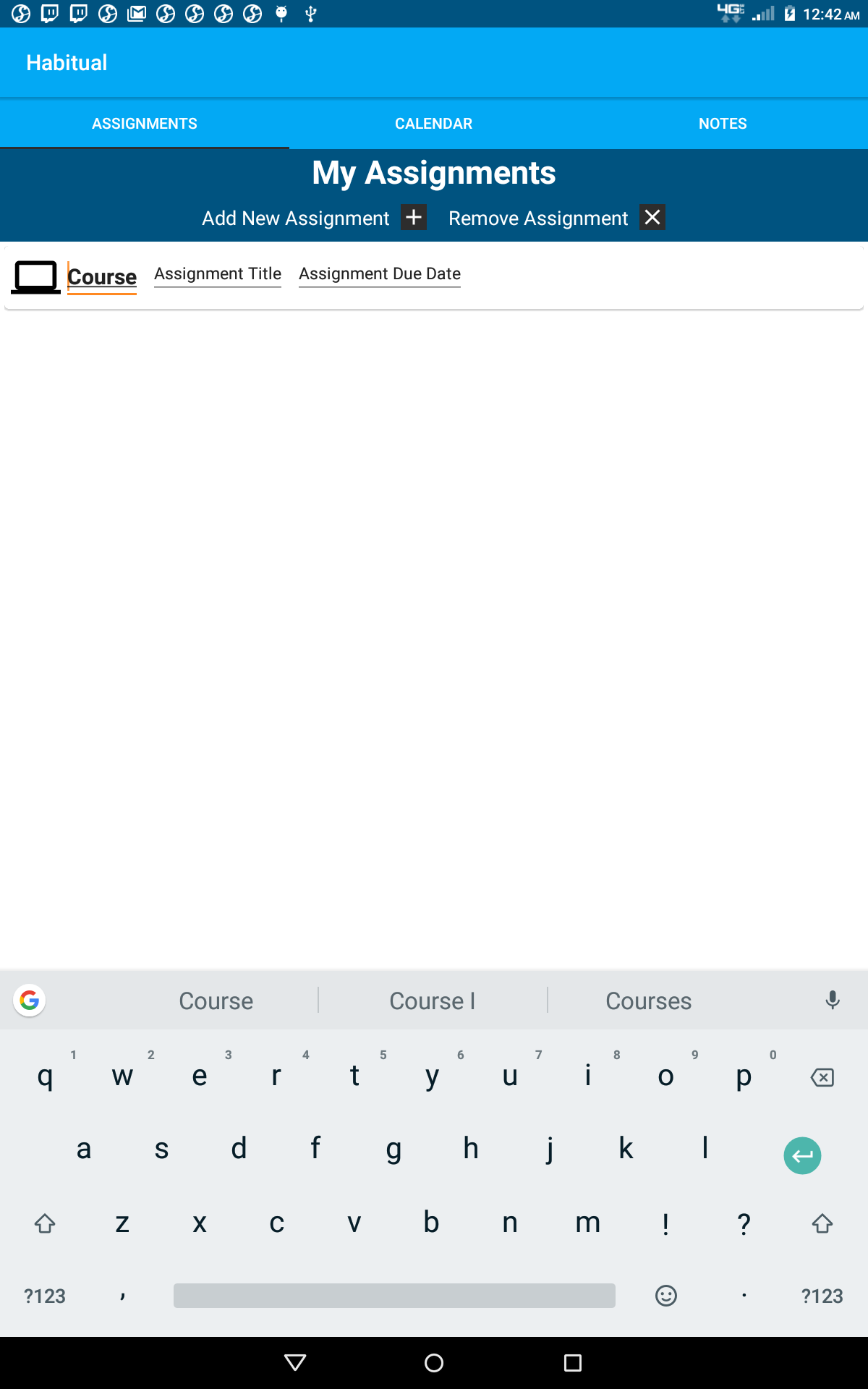
As far as development goes, I think what went well was learning how to lay everything out in a way that was much more appealing than before. Previously, I was using LinearLayouts that just stacked everything on top of each other. Now that I’ve switched to using RelativeLayouts for the buttons on the goal fragment, it looks much nicer. I’m planning to make the same sort of tweaks for the other two fragments as well. I think what didn’t go so well was learning how to send data between fragments and activities. I think in the future sprints, I will try to stick to using activities whenever possible because fragments are much trickier to work with.

# **Sprint #4**

Sprint Backlog

|  |  |  |  |
| --- | --- | --- | --- |
| **Story ID** | **Story / Task** | **Estimated**  **Hours** | **Actual**  **Hours** |
| 8 | User goals should be added to the list and viewable when adding through the “add new goal” button and dialog | 4 | 6 |
| 9 | Create a dialog for adding a new goal | 3 | 3 |
| 10 | Update UI to reflect business changes – goals are now “assignments”, progress is now “calendar”, and logging is now “notes”.1 | 1 | 1 |
| 11 | Update list to accept an icon, a course name, an assignment title, and a due date as the assignment information (collected from the custom dialog for adding a new assignment) | 4 | 3 |

## Review



This iteration, I focused more on the backend implementation regarding collecting user information from a custom dialog. I created a custom dialog that prompts the user for the assignment title, the assignment description, and the assignment due date. That information is then sent to the RecyclerView containing a list of Assignment objects and is stored there. As you can see in the first screenshot, each assignment in the list contains an image for the course, a course title, an assignment title, and an assignment due date. This information is collected from the user through the custom dialog. For the most part, I was able to complete everything that was planned for the sprint. There are a few bugs I need to iron out for next sprint but otherwise, the adding functionality of the assignments page is nearly complete.

## Retrospective

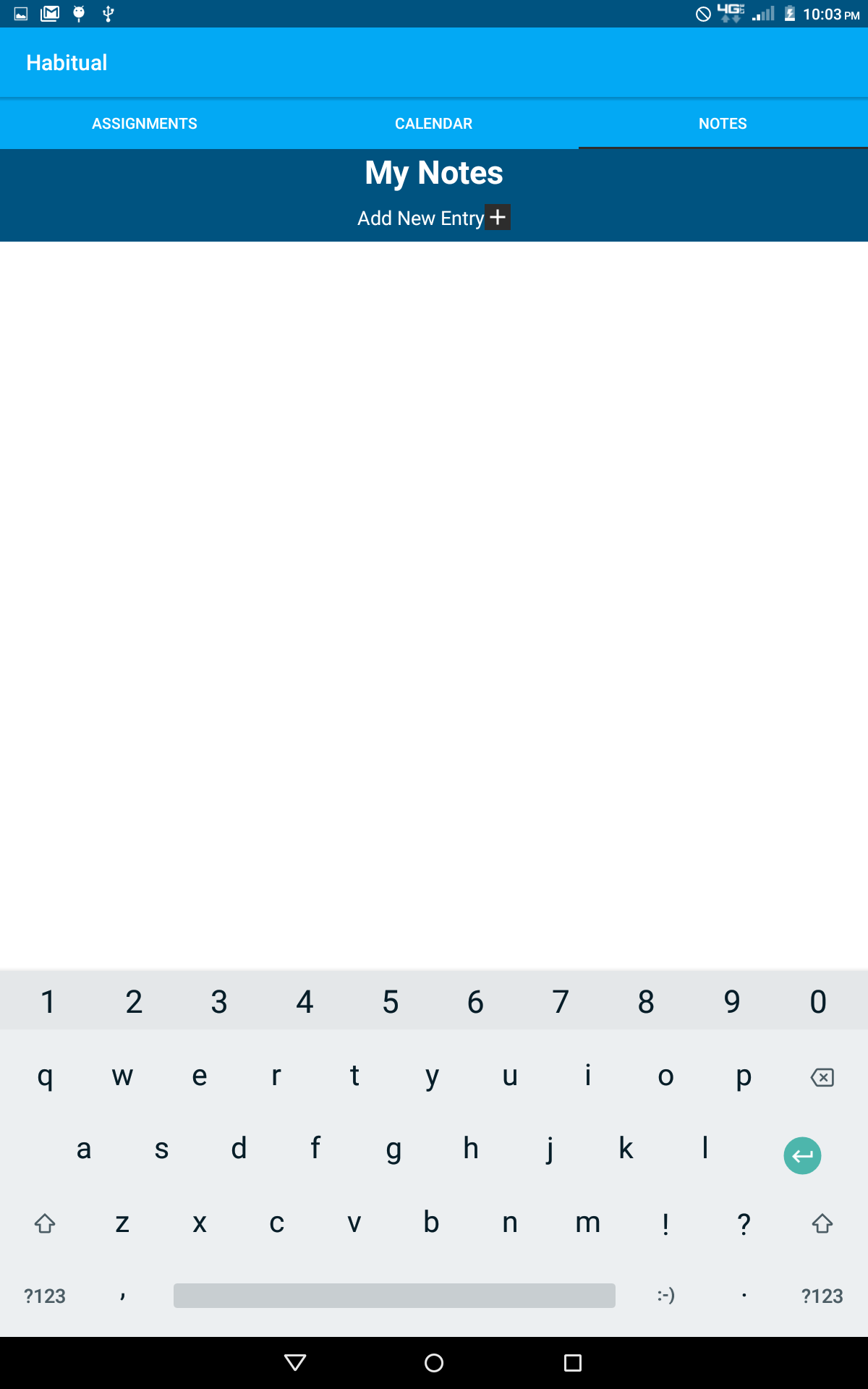
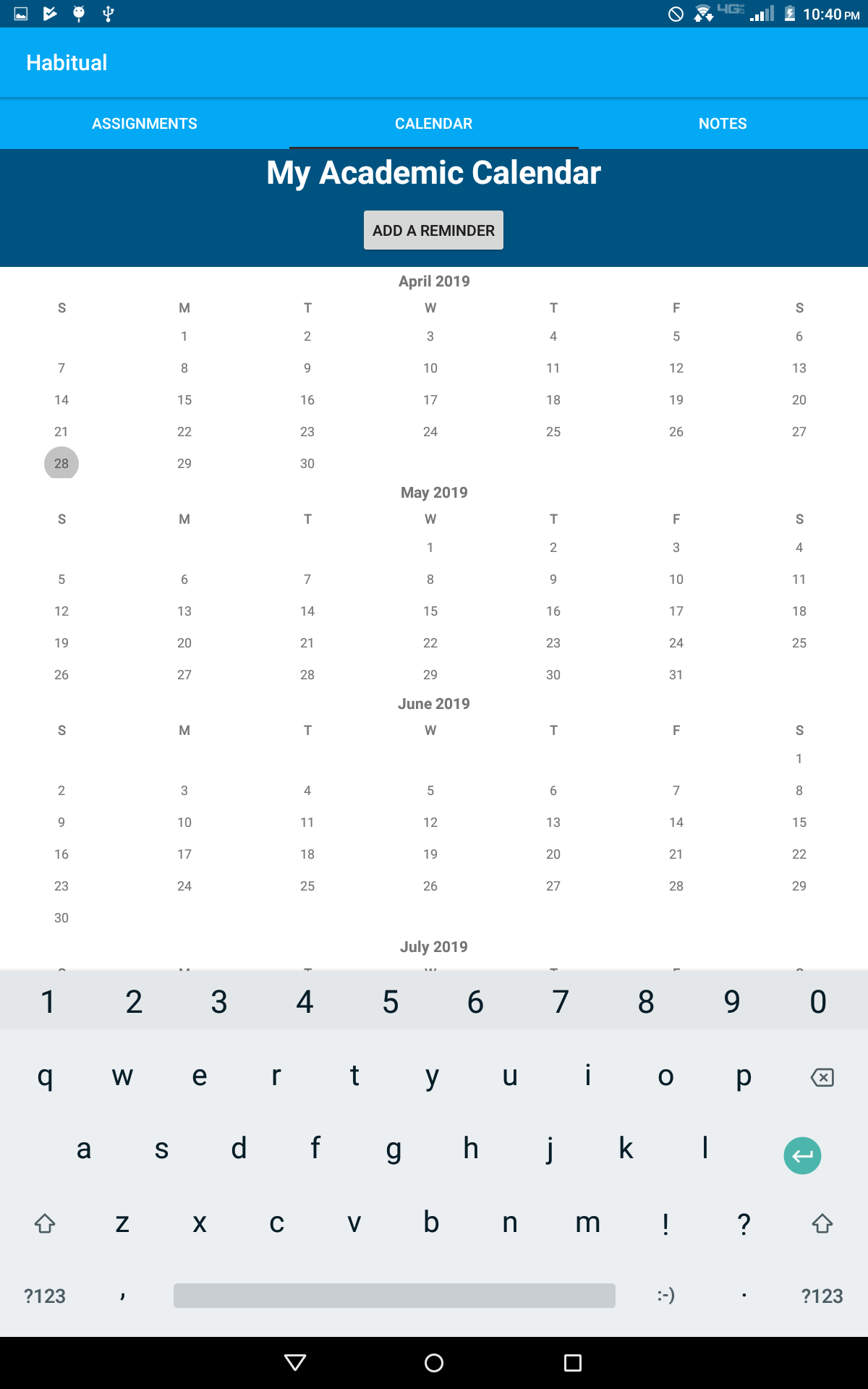
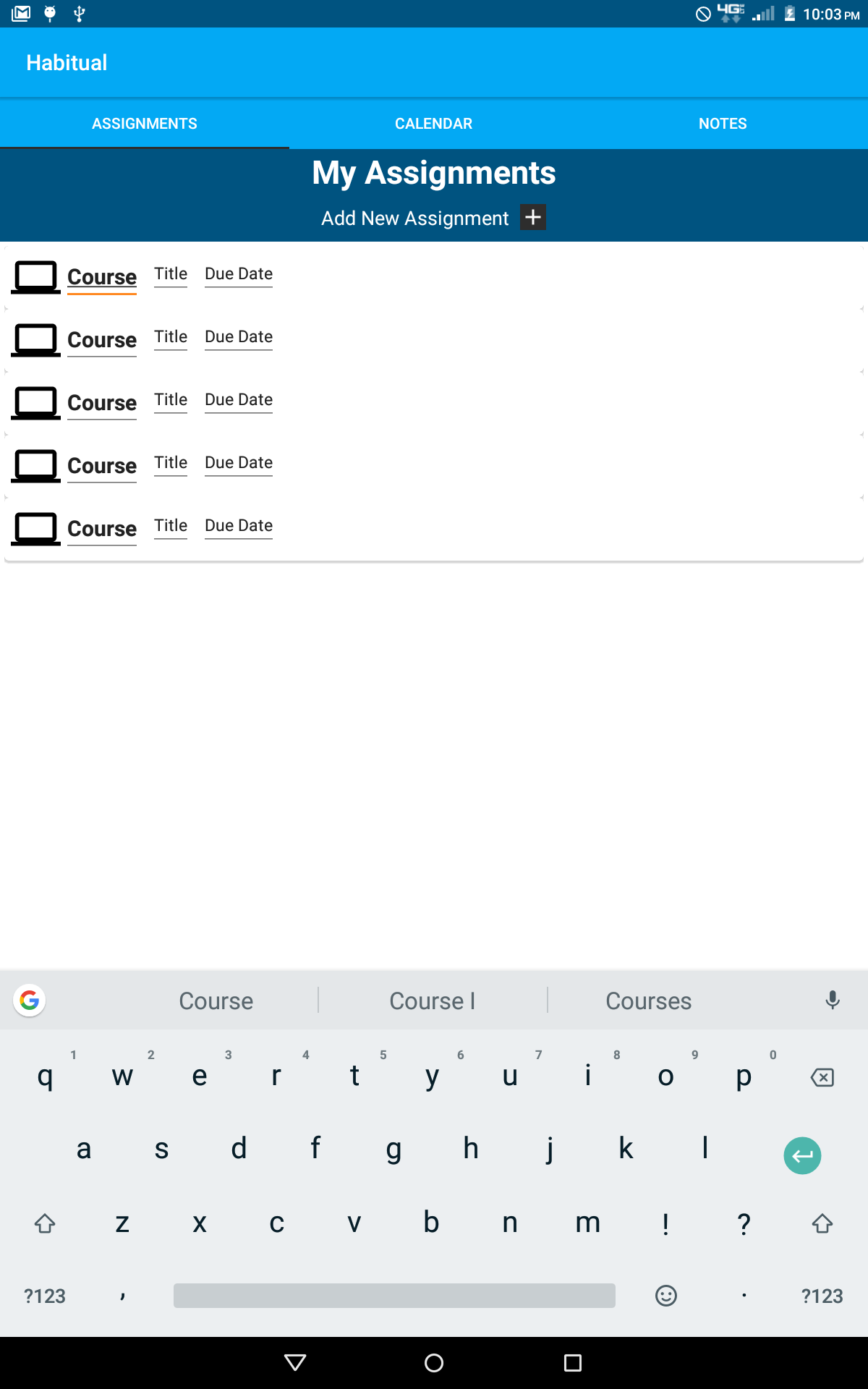
I think I was able to balance a good amount of front end and back end work this sprint. I am not necessarily planning to do this each sprint because back end implementation takes much longer than setting up the UI. I was able to follow some simple tutorials for adding custom dialogs and sending information between fragments. I’m planning to use these similar tutorials in the future to speed up the process. Next sprint, I plan to complete the assignments page by ironing out the adding assignment workflow to accept the user information and add it to the recycler view. I also plan to add the remove functionality. If all goes well, I will move on to the calendar section where the user can add/remove reminders.

# **Sprint #5**

Sprint Backlog

|  |  |  |  |
| --- | --- | --- | --- |
| **Story ID** | **Story / Task** | **Estimated**  **Hours** | **Actual**  **Hours** |
| Implement a button on the goal page to remove goals from the list and have them no longer appear on the goal page. User should also be able to select and remove multiple goals at once. | 4 | 1 | 5 |
| Add a button to add a reminder on the calendar page | 12 | 6 | 2 |
| Update UI to match across all 3 pages | 13 | 1 | 1 |
| Add functionality to add a note on the note page | 14 | 2 | 2 |

## Review



This sprint, I updated the UI to match across all 3 pages. More backend work was done to work on adding an assignment and adding notes. Both prompt the user for more information including titles, descriptions, and dates through dialogs which store that information in a recycler view. I was able to complete that as well as the adding a button to add a reminder, but the functionality for that will be moved to next sprint. I also decided to remove the remove functionality from the assignments and notes page until I can better figure out a way to do that. The way that I currently have the data stored, I haven’t found a way that will make it easy for the user to understand how to delete. I was hoping that the user could select multiple at the same time for deletion, but that doesn’t appear to work with the way that I have it set up. I will move this to a future sprint.

## Retrospective

I think I was able to be more productive this sprint because the assignment and note pages are very similar. I reproduced the same dialog and the same adapter for the note page as I did for the assignment page. I spent a lot of time trying to figure out a good way to delete the assignments/notes but I wasn’t able to find one that was straightforward. I removed it for the time being, but I may revisit it during the last sprint. For next sprint, I plan to add the backend functionality for adding a reminder and the finish up the note page (notes should appear in the recycler view).

# **Sprint #6**

Sprint Backlog

|  |  |  |  |
| --- | --- | --- | --- |
| **Story ID** | **Story / Task** | **Estimated**  **Hours** | **Actual**  **Hours** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Review

[Screenshots, etc go here. This is where you discuss the product, describing what was done this sprint (potentially shippable product increment) and what was planned for the sprint but was not done. ]

## Retrospective

[This is where you discuss the process. What went well (and are you planning to do more of that?) What didn’t go so well (and do you have a way to do less of that)? What changes are you planning to make in how you plan & carry out the next sprint?]