**Heading:** Sprint 3 Review, TP, TimeLords,  12 July 2016, #1, 28 July 2016

**Sprint Goal**: As a user I want to be able to add tasks and projects and edit their attributes as well as navigate between pages using the menu.

**Sprint 3 User stories**

1. (2) As a user I want to be able to **organize my tasks into projects** so that I can separate tasks and view them more easily.
2. (8) As a user I want to **add Task attributes, and easily edit them** to distinguish them.
3. (2) As a user I want to be able to **mark tasks as done** so that I know what tasks I have completed
4. (2) As a user I want to be **able to add notes to each task** so I can have a place to put other relevant information not provided by the pre-existing list of attributes
5. (3) As a user I want an user interface that allows me to **swipe from the menu page to other pages *flawlessly***, so I can navigate through the  application
6. (13) As a user I want the **app to REMEMBER my tasks when I exit the app** because otherwise it would be completely useless to even use this app.
7. (5)
8.
9.
10.
11.
12.
13.
14.
15.
16.
17.
18.
19.
20.
21.
22.
23.
24. As a user I want a “dump” page to hold all of my tasks that are not organized into a project so that I can create tasks not assigned to a specific project (the default project will be the dump)

25. (1) As a user I want to be able to attribute an estimated length to my tasks so I  (and later the app itself) can identify quick tasks easily
26. (1) As a user I want to be able to set priority levels for each of my tasks so I know what the most important tasks are.
27. (1) As a user I want to be able to “tag” my tasks so that later I can search and find all tasks with that tag.
28. (3) As a user I want a collapsible list of projects in my menu so that I can navigate the application easily
29. (8) As a User I want **access to a list of recently completed tasks** so I can reflect on my productivity.
30. (3) As a User I want to **access a list of completed projects** so I can look at what I’ve been doing with my life and either cry with joy or sadness
31. (5) As a user I want to be able to add projects from all pages so that I have to do less navigating
32. (5) As a user I want to be able to have a *satisfying* animation when I “complete” a task so that I can feel good about myself and my accomplishments.
33. (3) As a user I want to be able to delete projects in case I fux up
34. (5) As a user I want to be able to delete tasks to declutter lists.
35. (8) As a user I want a **sleek and intuitive graphic interface** so that I can use the app efficiently.

**Sprint 3 Task Listing**

~~i.~~ ~~**create project object**~~
~~ii.~~ ~~**be able to create instance of project via the floating action button**~~
~~iii.~~ ~~**be able to create instance of project via task edit page**~~
iv. **have functions to access projects and their fields in the database**
v. **display list of project instances in collapsible projects list**
vi. **have functions to edit projects and their fields**
~~vii.~~ ~~**create task object**~~
~~viii.~~ ~~**be able to create instance of task via the floating action button**~~
ix. **have functions to access projects and their fields in the database**
x. **display list of tasks in a given project**
xi. **have functions to edit projects and their fields**
~~xii.~~ ~~**have function that checks if task is completed**~~
~~xiii.~~ ~~**have function to mark task complete in instance**~~
xiv. **have function to link the status of the attribute in the instance with the check box in the UI**
~~xv.~~ ~~**create attribute in task object**~~
~~xvi.~~ ~~**be able to add notes via the task edit page**~~

**xvii.** **have function that takes the user input from task edit page and store it in task instance**

**xviii.** **have function that updates the notes section from the UI and store it in task instance**

~~**xix.**~~ ~~**create menu**~~

~~**xx.**~~ ~~**display dump, quick tasks, urgent tasks, and collapsible projects list in this menu**~~

**xxi.** **be able to access each of the above lists by clicking on the button from the menu**

**xxii.** **implement swiping function on all pages**

**xxiii.** Integrate backend with front end

xxiv. Possibly need in of editing one or both sides to make it work

xxv. Unit tests

**xxvi.** **i. create special dump instance**

**xxvii.** ~~**ii. make dump viewable via menu**~~

**xxviii.** **iii. make dump.java**

**xxix.** **iv. make dump accessible when clicked on**

~~**xxx.**~~ ~~**create duration field in task object**~~

**xxxi.** **create duration field in GUI**

**xxxii.** **make duration editable via GUI**

**xxxiii.** **store edited duration in task object**

**xxxiv.** ~~**i. add attribute to task object**~~

**xxxv.** **ii. add region in task GUI**

**xxxvi.** **iii. join it with database**

**xxxvii.** **i. add tab enumerated var. (so we would have limited number of available tags)**

**xxxviii.** **ii. Add to task GUI**

**xxxix.** **Iii. join it with database**

**xl.** ~~**i. add attribute to task object**~~

**xli.** **ii. add region in task GUI**

**xlii.** **iii. join it with database**

**xliii.** **i. add tab enumerated var. (so we would have limited number of available tags)**

**xliv.** **ii. Add to task GUI**

**xlv.** **Iii. join it with database**

**xlvi.** ~~**i. add collapsible projects list to menu**~~

**xlvii.** **ii. make method to pull from server side list of existing projects to display correct projects**

**xlviii.** **implement collapsible lists of task objects for each project (one “to-do” and one “completed”)**

xlix. create method to search through each task in a given project and store it in the collapsible “completed” list

l. create method to display the list of completed tasks

li. implement collapsible lists of project objects for each project (one “to-do” and one “completed”)

lii. create method to search through each project and store it in the collapsible “completed” list

liii. create method to display the list of completed projects

liv. I.make add task button available on all pages

lv. Ii.make this button functional on each page

lvi. I.create nice UI animations

lvii. i.add delete functionality to project class

lviii. Ii.remove tasks from UI

lix. Iii.deleting/update databaseClass functionality

lx. i.add delete functionality to task class

lxi. Ii.remove tasks from UI

lxii. Iii.deleting/update databaseOperations functionality

lxiii. i. decide on primary colors

lxiv. ii. decide on accent colors

lxv. iii. implement consistently on each page

lxvi. iv. tweak any remaining UI elements (buttons, fonts, etc) to look clean

---

~~1. create project object~~

~~2. be able to create instance of project via the floating action button~~

~~3. be able to create instance of project via task edit page~~

4. have functions to access projects and their fields in the database
5. display list of project instances in collapsible projects list
6. have functions to edit projects and their fields

~~7. create task object~~

~~8. be able to create instance of task via the floating action button~~

9. have functions to access projects and their fields in the database
10. display list of tasks in a given project
11. have functions to edit projects and their fields

~~12. have function that checks if task is completed~~

~~13. have function to mark task complete in instance~~

14. have function to link the status of the attribute in the instance with the check box in the UI

~~15. create attribute in task object~~

~~16. be able to add notes via the task edit page~~

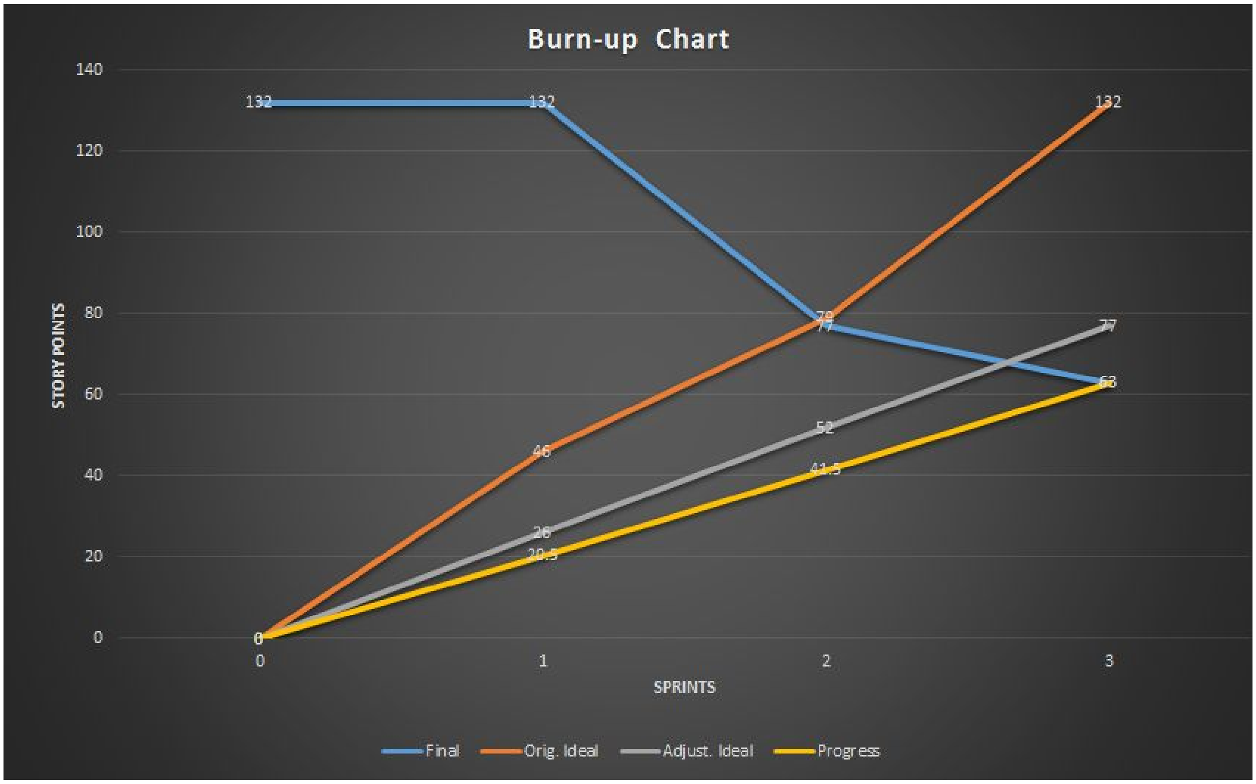
17. have function that takes the user input from task edit page and store it in task instance

18. have function that updates the notes section from the UI and store it in task instance
19. ~~create menu~~
20. ~~display dump, quick tasks, urgent tasks, and collapsible projects list in this menu~~
21. be able to access each of the above lists by clicking on the button from the menu
22. implement swiping function on all pages
23. Integrate backend with front end (db)
24. Possibly need in of editing one or both sides to make it work
25. Unit tests
26. i. create special dump instance
27. ~~ii. make dump viewable via menu~~
28. iii. make dump.java
29. iv. make dump accessible when clicked on
30. ~~create duration field in task object~~
31. create duration field in GUI
32. make duration editable via GUI
33. store edited duration in task object
34. ~~i. add attribute to task object~~
35. ii. add region in task GUI
36. iii. join it with database
37. i. add tab enumerated var. (so we would have limited number of available tags)
38. ii. Add to task GUI
39. Iii. join it with database
40. ~~i. add attribute to task object~~
41. ii. add region in task GUI
42. iii. join it with database
43. i. add tag enumerated var. (so we would have limited number of available tags)
44. ii. Add to task GUI
45. Iii. join it with database
46. ~~i. add collapsible projects list to menu~~
47. ii. make method to pull from server side list of existing projects to display correct projects
48. implement collapsible lists of task objects for each project (one “to-do” and one “completed”)
49. create method to search through each task in a given project and store it in the collapsible “completed” list
50. create method to display the list of completed tasks

51. create method to display the list of completed projects
52. I.make add task button available on all pages
53. Ii.make this button functional on each page
54. I.create nice UI animations
55. i.add delete functionality to project class
56. Ii.remove tasks from UI
57. Iii.deleting/update databaseClass functionality
58. i.add delete functionality to task class
59. Ii.remove tasks from UI
60. Iii.deleting/update databaseOperations functionality
61. i. decide on primary colors
62. ii. decide on accent colors
63. iii. implement consistently on each page
64. iv. tweak any remaining UI elements (buttons, fonts, etc) to look clean

---

## Burn up Chart



## Initial SCRUM Board

**Tasks Completed:** (green=completed, orange=in progress)

**Team Roles and Task Assignment**

Julia Moffitt (PO)

- **Task 4 add project attribute to task (.5)**
-
-

Sergiy Ravnyago (SM)

- **Add to task GUI (2)(task)**
-

Michaela Molina

- **Task 1 create special dump instance (2)**
-

Jeff Wagenseller

- **Task 2 implement file to save user data (10)**
-

Brian Houts

-