

- ★ Throughout the last time we delivered deliverable 2, we have individually made changes to the code. On 5/30/15, we scheduled to get together at 7:30pm over google hangouts to make sure everything is together, any code smells, and making sure all user stories and business rules were handled correctly.
- ★ We changed all the test naming conventions from “shouldBlahBlahBlah”, into “test BlahBlahBlah” method names because we preferred that as our naming convention.
- ★ We changed most of the JUnit Tests to reflect the black box testing style that we covered in class in order to make sure the code did not break. In order to check these tests we needed to use edge cases. The JUnit Tests were given very specific headers in order to explain to others what is happening in that one method.
- ★ For the classes, we also created static final constants in order to get rid of the magic numbers and provide a less smelly way to use constant numbers.
- ★ We also updated the UML design since it wasn’t reflecting what our model code is now.
- ★ Listed below is the refactoring we made to our code to make it better coding technique, much less smelly (to the best of our knowledge), more efficient, and nicer to look through
- ★ Lots and lots of refactoring:
  - **Admin** class, the method getVolunteer() was cleaned up and iterator was removed and used a do for loop instead to minimize the number of lines.
  - **CerealTest** class, the park names of all the jobs we created have different job titles; other than “tacoma wa” that it was before.
  - **Job** class, the address variable was removed because we didn’t need/use that detail and all of the getters and setters and anything pertaining to address were removed accordingly.
  - **JobTest** class, since the address variable was removed, all the jobs being created had to not include the address. Plus the shouldCompareDifferentAddresses method was removed. A lot of test methods were created to ensure business rules were being enforced and allowing full code coverage.
  - **ParkManager** class, cleaned up useless code, minimized a bit of code redundancy, and added a new method called isFileFound().
  - **ParkmanagerTest** class, since the address variable was taken out, all the jobs needed to change accordingly.
  - **Volunteer** class, took out useless code and also split up a method and added a new method called isSameDay() because it was checking for too much. We split up the method which makes it easier to understand and avoids code smells.
  - **VolunteerTest** class, fixed the created jobs because the address variable was removed.

- **AdminGui** class, changed import and removed useless code because we had unused lines of code and renamed variables.
- **JobGui** class, added a variable to make the code easier to read and just makes the class nice and clean looking.
- **LoginGui** class, changed the if and else if statements to be more specific to each of the users. Makes the code easier to understand what the variable “access” is trying to compare to and what it will be doing.
- **ParkManagerGui** class, since the LoginGui class was changed we needed to change it accordingly since it will be receiving a method correlating with its specified user. Also, took out a bunch of useless methods and renamed for easier readability. There was also a lot of breaking down methods to make them do specific things and make the code easier to read; like adding a new method called getJobs() to allow the Park Manager to see what jobs they have submitted. Also adding a new method called selectJob() which is the step before grabbing all the volunteers for that specific job.
- **VolunteerGui** class, just like the ParkManagerGui class, since the LoginGui class was changed, we needed to update the connecting classes, so we had to change methods' parameters to correlate with what the LoginGui is passing. We also took out a bunch of useless code and modified a few if statements to make the class cleaner and easier to read. In addition to make code easier to read and avoiding code smells, we created a method call currentJobList() which returns the current list of jobs.