Project State for Deliverable 3

Group 1: Robert, Lachezar, Duong, Bethany

User Stories:

1. As a Park Manager I want to submit a new job.
   Extend code is written: Fully implemented.
   What is tested: The UrbanParkCalendar class throws exceptions if any business rule regarding adding a job would be violated, which includes: Attempting to add a job if the Calendar is already full, adding a duplicate job, adding a job with too many volunteers, adding a job that would have already happened, adding a job that is too far into the future, adding a job that is more than 48 hours in length, adding a job that starts after it ends and adding a job to a week that is already full.
   Verified via user interface: Yes. Users can verify the new job exists with User stories 2, 3, 8 and 9. Any feedback regarding failure is done immediately to the Park Manager.
2. As a Park Manager I want to delete a job.
   Extend code is written: Fully Implemented.
   What is tested: Testing that a job was removed from the Calendar is tested in Calendar Test.
   Verified via user interface: Yes. Users can verify the job is gone by either trying to delete it again or attempting to edit.
3. As a Park Manager I want to edit the details of a job.
   Extend code is written: Fully implemented.
   What is tested: Tests that a job can be edited with no volunteers, and can't be edited with some volunteers. Testing that editing works properly is tested in Calendar Test. Additional tests are done after new details are entered but before the job is actually edited, such as making sure the job does not become a duplicate or a job that already exists, or editing the dates to violate a different business rule.
   Verified via user interface: Yes. Users can verify that the job is changed by viewing it again.
4. As an Volunteer, I want to view a summary of all upcoming jobs.
   Extend code is written: Fully implemented. Volunteers are able to see all upcoming jobs, but not past jobs.
   What is tested: Jobs are checked against the current time before being returned to ensure that past jobs are not displayed. JUnit Tests in UrbanParkCalendar verify this.
   Verified via user interface: Yes. The volunteer can see this through menu commands.
5. As an Volunteer, I want to view details of a selected upcoming job.
   Extend code is written: Fully implemented.
   What is tested: Not tested. This is verified through the user interface, since this simply uses a getter to get the user's list of jobs.
   Verified via user interface: Yes. Volunteers can check this function by their menu.

6. As a Volunteer I want to volunteer for a job.
   Extend code is written: Fully implemented, Volunteers can sign up for appropriate jobs. Volunteers can only sign up for jobs that do not break any business rules.
   What is tested: All rules are tested, such that a volunteer is able to successfully sign up for a job as long as the job does not interfere with any of the other jobs volunteers have, and is not a past job.
   Verified via user interface: Yes. Volunteers can see this by looking at the details of a job.
7. As a Volunteer I want to view the jobs I am signed up for.
   Extend code is written: Fully implemented. Volunteers can see all jobs they are signed up for.
   What is tested: Volunteers keep track of the jobs they are signed up for.
   Verified via user interface: Yes. This is one of the functions in their menu.
8. As a Park Manager I want to view a summary of all upcoming jobs in the parks that I manage.
   Extend code is written: Fully Implemented
   What is tested: Similar to User Story 4, except that only jobs that appear in the Park Manager's park(s) are returned.
   Verified via user interface: Yes. Park Managers can test this functionality through their menu.
9. As a Park Manager I want to view the Volunteers for a job in the parks that I manage.
   Extend code is written: Fully Implemented
   What is tested: Only parks managed by that particular manager are checked for jobs and Volunteers.
   Verified via user interface: Yes. Park Managers can view this in their menu.
10. As an Urban Parks staff member, I want to search volunteers by last name.
    Extend code is written: Fully implemented.
    What is tested: This simply gets volunteers from the calendar, and tests in Calendar and Job test that volunteers are working properly in these classes.
    Verified via user interface: Yes. Urban Parks Staff Members can check this function from the menu.
11. As an Urban Parks staff member, I want to view a summary of all upcoming jobs.
    Extend code is written: Fully Implemented
    What is tested: Tested in the same fashion as User Story 4.
    Verified via user interface: Yes. There is a menu option for Staff members to view all upcoming jobs.
12. As an Urban Parks staff member, I want to view details of a selected upcoming job
    Extend code is written: Fully implemented. An Urban Parks Staff Member can view the specific details of any job from the master list.
    What is tested: Nothing. Since this works by Staff member's simply get a list of upcoming jobs to view a job, we did not test simple getter methods.
    Verified via user interface: Yes. The park manager can select any job from the displayed list.

Business Rules:

1. A job may not be added if the total number of pending jobs is currently 30.
   Extend code is written: Fully implemented. Park Managers cannot create the job if the Calendar is too full.
   What is tested: Tests in the calendar catch an exception thrown when too many jobs are added to the calendar, also tests in Job that hasMaxJobs works correctly.
   Verified via user interface: Yes. A Park Manager will be warned if there are too many jobs currently in the Calendar.
2. A job may not be added if the total number of pending jobs during that week (3 days on either side of the job days) is currently 5. In other words, during any consecutive 7 day period there can be no more than 5 jobs.
   Extend code is written:Fully Implemented
   What is tested: testCheckForRoomThatWeek tests that the dates 3 days before and after are calculated and then the calendar is iterated over to check how many jobs are already in that time span. If it's 5 or more, then the Calendar rejects the new job with an exception.
   Verified via user interface: Yes. Park Managers can attempt to create 5 jobs on the same day to easily verify this.
3. A Volunteer may not sign up for a work category on a job if the maximum number of Volunteers for that work category has already been reached.
   Extend code is written: Fully Implemented
   What is tested: Tests in volunteer check that a volunteer cannot exced the maximum number of volunteers in any work category.
   Verified via user interface: Yes, a volunteer will be notified via the user interface that the job is full.
4. A job may not be scheduled that lasts more than two days.
   Extend code is written: Fully Implemented
   What is tested: We have tests to check that the duration is not longer than two days and that the appropriate exception is thrown if it is.
   Verified via user interface: Yes. The user will be notified via the user interface that the job is too long.
5. A job may not be added that is in the past or more than three months in the future.
   Extend code is written: Fully Implemented
   What is tested: The requested date is checked against dates that have already occurred and dates 90 days from now. If it fails either check, the job is not created and an appropriate exception is thrown.
   Verified via user interface: Yes. The user trying to add a job will be notified that the job is too far in the future.
6. A Volunteer may not sign up for a job that has passed.
   Extend code is written: Fully Implemented
   What is tested: Nothing. This is enforced via the interface.

Verified via user interface: No, a Volunteer does not have access to past jobs, so they will never be able to sign up for a past job.

7. A Volunteer may not sign up for two jobs on the same day.
   Extend code is written: Fully Implemented
   What is tested: There are tests to make sure that jobs that conflict will not let a volunteer sign up for a job.
   Verified via user interface: Yes, the user will be notified that the job they are signing up for interferes with another.
8. A Park Manager can create jobs only for those parks that he/she manages.
   Extend code is written: Fully Implemented. Park Managers only have access to their own parks.
   What is tested: Nothing. This is enforced via the interface.
   Verified via user interface: Yes. ParkManagers can only see their own parks and are asked to specify which park they would like to create a job in before going any farther.