**Unit Test Skeleton**

public class Test{

public void testFunctionalies(){

//tests the game functionalities

}

public void testPayPlatform(){

//tests that the pay platform is connected properly

}

public void testTwitter(){

//tests twitter plug in

}

public void testInstagram(){

//test Instagram plug in

}

public void testFacebook(){

//test facebook plug in

}

public void testParentInfo(){

//test that all inputted info is saved properly in the database

}

public void testStudentInfo(){

//test that all inputted info is saved properly in the database

}

public void testGameStats(){

//test that all game stats are saved properly in the database

}

public void testStudentRegistration(){

//tests that students can register

}

public void testParentRegistration(){

///tests that parents can register

}

public void testPhoto(){

//tests that photos can be submitted

}

public void testReview(){

//tests reviews can be submitted

}

public void testParkDescription(){

//tests that park descriptions can be submitted

}

public void testOfficialReport(){

//tests the automatic generation of an official report after the game is played

}

public void testParentReports(){

//tests the automatic generation of a parental report after the game is played

}

}

**Integration Testing Skeleton**

public class Test{

public void testGame(){

//combines and test functionalities() units

}

public void testDonations(){

//combines and test payPlatform() unit

}

public void testSocialMedia(){

//combines and test twitter() , instagram(), and facebook() units

}

public void testDatabase(){

//combines and test parentInfo(), studentInfo(), gameStats() units

}

public void testRegistration(){

//combines and test studentREgistration(), parentRegistration() units

}

public void testSubmissions(){

//combines and test photos(), reviews(), parkDescriptions() units

}

public void testReports(){

//combines and test officalReports(), parentReports() units

}

}

**Regression Test Procedure**

When changes are integrated into the code we will follow the following procedure to ensure the code does not create a bug. We will prioritize certain tests due to their importance and to save time and money. If any of these tests present a bug in the system, stop and begin a top down approach to find it, for example, try the integration tests, and if they do not expose the bug, then go all the way down to unit tests to isolate the issue.

1. Re-run testDonations()
2. Re-run testDatabase()
3. Re-run testGame()
4. Re-run testRegistration()

These four pieces are the most critical to the system success.

**System Test Procedure**

Our system test procedure can be completed in levels; unit testing, integration testing, and system testing. Our first order of business will be to follow the unit test skeleton outlined above to ensure that all small functionalities are working. We will then test all the bigger pieces by using integration testing; we will combine all our units and test each class and ensure that we are getting correct outcomes. Our final step will be to test the whole system together. We will use all the classes outlined and ensure they’re working as well. If there are changes or errors at any steps we can refer to the prioritization discussed in the regression testing procedure to decide how we will find the bug.