3.1 Functional Requirements
3.1.1 User Interface

3.1.1.1 Admin Access (Terrell)
Power Play must allow users with the proper authorization to approve, deactivate/reactivate and edit user accounts. Power Play must provide the ability to input/edit user information via the Account Management page.

A. Account Deactivation/Reactivation
An account must already exist see requirement 3.1.3.1
The following functional requirements must be met:
1. Provide the capability for administrator users to deactivate player accounts in Power Play. This removes player access while keeping player data
2. Provide the capability for administrator users to reactivate player accounts in Power Play. This reinstates player access while keeping player data

B. Account Password Reset
An account must already exist see requirement 3.1.3.1. The admin user will be able to manually reset the password for Power Play players that have forgotten the password to their account.
1. Provide the capability for administrators to reset the password of player accounts
2. The default password to be used is:
   a. dominion

C. New Account Approval
1. Provide the capability for administrators to approve new accounts created by players

D. Account Permission Modification
1. Provide the capability for administrators to change access levels for player accounts

3.1.1.2 Authentication (Addy)
Power Play must allow users to properly authenticate prior to accessing game functionality. Power Play must ensure users are properly authenticated prior to allowing players to access profile, player data, and game features. The system must allow authorized users to:

A. Username and Password Verification.
   1. Provide the functionality of verifying username and password with a database
B. Provide security for different roles of a user (Administrator, Analyst, Player)
   1. Provide the functionality of verifying specific views and permissions that are
      allowed based on user role.
   2. Restrict certain functions to authorized personnel only

3.1.1.3 Access Control (Addy)
The system must provide three user roles with subsequent permission sets detailed in
the sections below. All users have player role permissions, analysts have player and analyst
role permissions, and an admin has all available permissions within the system. Roles can only
be granted by a user with the Administrator role.
   A. Player Access Role
      1. Provide registered user access to the gameplay functionality.
   B. Admin Access Role
      1. Provide Admins access to user accounts if the registered user has Administrator
         permissions
   C. Analyst Access Role
      1. Provide registered user access to Analyst features including reporting dashboard
         and game history if the user has Analyst permissions.

3.1.1.4 Player Sign-Up (Addy)
The Player must be provided with the ability to register a new account.
   A. Account Creation
      The following requirements must be met in the user creation screen, a form will be
displayed requiring the information specified below. When the information is submitted it
will confirm that the account was completed and require confirmation via email from
approved ODU domains only.
      1. Desired username
      2. Desired password
      3. Email address (from approved ODU domain only)
      4. Password confirmation
      5. Player icon

3.1.1.5 Account Controls (Ryan)
This is a page that provides a logged in user with the ability to edit their profile and
information. This page will be populated with data from the users table of the database.
The following functional requirements must be met:
   A. View Profile
      The user must have the ability to view their profile information.
      Information includes the following:
1. Previous Gameplay data
   Gameplay history displayed from the database. Includes averages and statistical
data of player performance.
   a. Rounds played
   b. Total revenue
   c. Environmental impact
   d. Energy availability.
   e. Research completed
   f. Scenarios completed

2. Change Password
   The user must have the ability to change their password. A form will be provided
   with a password and password confirmation. The form will test for minimum
   complexity standards to ensure security.

3. Delete Profile
   The user must have the ability to delete their profile and associated data if desired
   through this form.

3.1.1.6 Reporting
   Power Play requires a page that presents data visualizations pulled from the database to
   the player for research and decision making purposes. The following functional
   requirements must be met:
   A. Projected Gameplay Graphed
      The user must be able to view data visualizations of current decision impacts
   1. View Monetary Resources
      The user must have the ability to view the following data:
      a. Revenue
      b. Available cash
      c. Amount of cash used
   2. View Environmental Impact
      The user must have the ability to view the amount of carbon emissions produced
      by energy production.
   3. View Energy Production
      The user must have the ability to view the amount of energy produced.
   4. View Supply/Demand of Energy Technologies
      The user must have the ability to view current availability of resources and
      demand from consumers.

3.1.2 Database Requirements - Dave
Power Play requires a database to store data that will be sent to the simulation engine as well as data that will be required to provide reporting to users.

1. The data will be stored in a MySQL database.
2. The database will use the Innodb storage engine.
3. The database will use UTF8mb4_unicode_ci character encoding.

3.1.2.1 Tables

1. User Table
   a. The database will have a table named users.
   b. The users table will have a primary key with the field name of user_id.
   c. The primary key will be stored as an integer.
   d. The primary key will not be null.
   e. The primary key will be set to auto increment.
   f. The user table will have a field for username.
   g. The username will be stored as a varchar.
   h. The user table will have a field for password.
   i. The password will be stored as a varchar.
   j. The password will be hashed before updating the database.
   k. The user table will have a field for a user’s email address.
   l. The email field will be stored as a varchar.

2. User Portfolio Table
   a. The database will have a table named user_portfolio.
   b. The user_portfolio will have a primary key with the field name of portfolio_id.
   c. The primary key will be stored as an integer.
   d. The primary key will not be null.
   e. The primary key will be set to auto increment.
   f. The user_portfolio table will have a foreign key with the field name of user_id.
   g. The user_portfolio table will have a field named hours_played.
   h. The hours_played field will be stored as a float.

3. User Roles Table
   a. The database will have a table named user_roles.
   b. The user_roles table will have a primary key with the name role_id.
   c. The role_id field will be stored as an integer.
   d. The role_id field will not be null.
   e. The role_id field will be set to auto_increment.
   f. The user_roles table will have a foreign key with the name user_id.
   g. The user_id field will be stored as an integer.
   h. The permissions field will be stored as an integer.

4. Scenarios Table
a. The database will have a table named scenarios.
b. The scenarios table will have a primary key named id.
c. The id field will be stored as an integer.
d. The scenarios table will have a field named name.
e. The name field will be stored as a varchar.

5. Scenario Assignees Table
a. The database will have a table named scenarios_assignees.
b. The scenarios_assignees table will have a primary key named id.
c. The id field will be stored as an integer.
d. The id field will be set to auto increment.
e. The scenarios_assignees table will have a foreign key named scenario_id.
f. The scenario_id field will be stored as an integer.
g. The scenarios_assignees table will have a foreign key named player_id.
h. The player_id field will be stored as an integer.

6. Zones Table
a. The database will have a table named zones.
b. The zones table will have a primary key with the name zone_id.
c. The primary key will be stored as an integer.
d. The primary key will auto increment.
e. The zones table will have a field with the name zone_name.
f. The zone_name field will be stored as a varchar.

7. Zone Data Table
a. The database will have a table named zone_data.
b. The zone_data table will have a primary key with the name zone_id.
c. The primary key will be stored as an integer and set to autoincrement.
d. The zone_data table will have a foreign key with the name scenario_id.
e. The scenario_id field will be stored as an integer.
f. The zone_data table will have a foreign key with the name zone.
g. The zone field will be stored as an integer.
h. The zone_data table will have a field named date.
i. The date field will be stored as a UNIX timestamp.
j. The zone_data will have a field named load_type.
k. The load_type field will be stored as a varchar.
l. The zone_data will have a field named quantity.
m. The quantity field will be stored as an integer.

8. Technology Cost Table
a. The database will have a table named technology_cost.
b. The technology_cost table will have a primary key named id.
c. The id field will be stored as an integer.
d. The technology_cost table will have a foreign key named scenario_id.
e. The scenario_id field will be stored as an integer.
f. The technology_cost table will have a field named year.
g. The year field will be stored as an integer.
h. The technology_cost table will have a field named technology.
i. The technology field will be stored as a string.
j. The technology_cost table will have a field named fixed_cost.
k. The fixed_cost field will be stored as an integer.
l. The technology_cost table will have a field named variable_cost.
m. The variable_cost will be stored as an integer.
n. The technology_cost table will have a field min_capacity.
o. The min_capacity field will be stored as an integer.
p. The technology_cost table will have a field max_capacity.
q. The max_capacity field will be stored as an integer.
r. The technology_cost table will have a field named carbon_dioxide.
s. The carbon_dioxide field will be stored as an integer.

9. Fuel Forecast Table
   a. The database will have a table named fuel_forecast.
   b. The fuel_forecast table will have a primary_key named id.
   c. The id field will be stored as an integer.
   d. The id field will be set to auto increment.
   e. The fuel_forecast table will have a foreign key named scenario_id.
   f. The scenario_id field will be stored as an integer.
   g. The fuel_forecast table will have a field named year.
   h. The year field will be stored as an integer.
   i. The fuel_forecast table will have a field fuel.
   j. The fuel field will be stored as a varchar.
   k. The fuel_forecast table will have a field cost.
   l. The cost field will be stored as a float.

10. Solar Potential Table
    a. The database will have a table named solar_potential.
    b. The solar_potential table will have a primary key named id.
    c. The id field will be stored as an integer.
    d. The id field will be set to auto increment.
    e. The solar_potential table will have a foreign key named scenario_id.
    f. The scenario_id field will be stored as an integer.
    g. The solar_potential table will have a foreign key named zone_id.
    h. The zone_id field will be stored as an integer.
    i. The solar_potential table will have a field named watts.
    j. The watts field will be stored as an integer.
11. Wind Potential Table
   a. The database will have a table named wind_potential.
   b. The wind_potential table will have a primary key named id.
   c. The id field will be stored as an integer.
   d. The id field will be set to auto increment.
   e. The wind_potential table will have a foreign key named scenario_id.
   f. The scenario_id field will be stored as an integer.
   g. The wind_potential table will have a foreign key named zone_id.
   h. The zone_id field will be stored as an integer.
   i. The solar_potential table will have a field named wind_speed.
   j. The wind_speed field will be stored as an integer.

12. Reliability Table
   a. The database will have a table named reliability.
   b. The reliability table will have a primary key named id.
   c. The id field will be stored as an integer.
   d. The id field will be set to auto increment.
   e. The reliability table will have a foreign key named player_id.
   f. The player_id field will be stored as an integer.
   g. The reliability table will have a foreign key named zone_id.
   h. The zone_id field will be stored as an integer.
   i. The reliability table will have a field month.
   j. The month field will be stored as an integer.
   k. The reliability table will have a field year.
   l. The year field will be stored as an integer.
   m. The reliability table will have a type field.
   n. The type field will be stored as a varchar.
   o. The reliability table will have a field named reliability.
   p. The reliability field will be stored as a float.

13. Asset Utilization Table
   a. The database will have a table named asset_utilization.
   b. The asset_utilization table will have a primary key named id.
   c. The id field will be stored as an integer.
   d. The id field will be set to auto increment.
   e. The asset_utilization will have a foreign key named player_id.
   f. The player_id field will be stored as an integer.
   g. The asset_utilization table will have a foreign key named zone_id.
   h. The zone_id field will be stored as an integer.
   i. The asset_utilization will have a field named month.
   j. The month field will be stored as an integer.
   k. The asset_utilization will have a field named year.
   l. The year field will be stored as an integer.
m. The asset_utilization will have a field named capacity_type.

n. The capacity_type field will be stored as a string.

 o. The asset_utilization table will have a field capacity_utilization.

 p. The capacity_utilization field will be stored as a float.

14. Economic Performance Table
   a. The database will have a table named economic_performance.
   b. The economic_performance table will have a primary key named id.
   c. The id field will be stored as an integer.
   d. The economic_performance table will have a foreign key named player_id.
   e. The player_id field will be stored as an integer.
   f. The economic_performance table will have a field named tech_type.
   g. The tech_type field will be stored as a string.
   h. The economic_performance table will have a field named cost.
   i. The cost field will be stored as a float.
   j. The economic_performance table will have a field named ror.
   k. The ror field will be stored as a float.

15. Environmental Performance Table
   a. The database will have a table named enviro_performance.
   b. The enviro_performance table will have a primary key named id.
   c. The id field will be stored as an integer.
   d. The id field will be set to auto increment.
   e. The enviro_performance table will have a foreign key named player_id.
   f. The player_id field will be stored as an integer.
   g. The enviro_performance table will have a field named carbon_dioxide.
   h. The carbon_dioxide field will be stored as a float.
   i. The enviro_performance table will have a field named nitrous_oxide.
   j. The nitrous_oxide field will be stored as a float.
   k. The enviro_performance table will have a field named carbon_monoxide.
   l. The carbon_monoxide field will be stored as a float.
   m. The enviro_performance table will have a field named sulfur_dioxide.
   n. The sulfur_dioxide field will be stored as a float.

16. Plants Table
    a. The database will have a table named plants.
    b. The plants table will have a primary key named plant_id.
    c. The plant_id will be stored as an integer.
    d. The plant_id will be set to auto increment.
    e. The plants table will have a foreign key named zone_id.
    f. The zone_id field will be stored as an integer.
    g. The plants table will have a foreign key named player_id.
    h. The player_id field will be stored as an integer.
i. The plants table will have a field named name.
j. The name field will be stored as a varchar.
k. The plants table will have field named type.
l. The type field will be stored as a varchar.
m. The plants table will have a field named capacity.
n. The capacity field will be stored as a float.
o. The plants table will have a field named production_commit.
p. The production_commit will be stored as a float.
q. The plants table will have a field named operational_cost.
r. The operational_cost field will be stored as a float.
s. The plants table will have a field upgrade_level.
t. The upgrade_level field will be stored as an integer.
u. The plants table will have a field named online_status.
v. The online_status field will be stored as a boolean.
w. The plants table will have a field named pollution_level.
x. The pollution_level field will be stored as a float.
y. The plants table will have a field named efficiency.
z. The efficiency field will be stored as a float.

17. Plant Metadata Table
   a. The database will have a table named plant_metadata.
   b. The plant_metadata table will have a primary key named meta_id.
   c. The meta_id field will be stored as an integer.
   d. The meta_id field will be set to auto increment.
   e. The plant_metadata table will have a foreign key named plant_id.
   f. The plant_id field will be stored as an integer.
   g. The plant_metadata table will have a field named meta_value.
   h. The meta_value field will be stored as a longtext.