Front Matter:

The following requirements are for my senior project, which is a new and improved version of the budget game used by the CSU Center for Personal Financial Management (CPFM) to help incoming freshmen get a better idea of what their lives will be like after graduation in terms of their personal finances and to help them become more interested in managing their finances well both while in college and after college. Specifically, this project will be a digital version of the budget game that will use a web scraper and a database to find and store up-to-date salary and cost data to be used in the game. I plan to use Oracle SQL and APEX for the database, Oracle APEX for the game app (which can connect easily to the database because both the app and the database will be stored in the same Oracle APEX Workspace), and Python for the web scraper.

Requirement #: 1 (Whole Project)  
Requirement Type: Security

Description: We will not store any personal data in this project because doing so would deviate from the original intent.

Rationale: There is no need to store personal information for this project, so I want to clarify that upfront here.  
Fit Criterion: Make sure that all data stored is obtained from publicly-available sources and that there is no personal user information in the project database.

Priority: Top priority

Dependencies: None. This requirement should be done first and upheld throughout the entire project.

Requirement #: 2 (Create Database Schema and Tables)  
Requirement Type: Functional

Description: The database schema and tables need to effectively store all the data required to make this project work, which includes average salary data for the different jobs available in the simulation game as well as all the prices and costs for things like places to live, cars, clothes, technology, and other expenses that would be included in the simulation game. Should also be compatible with data insertion from .csv files.

Rationale: If the data is not easy to access or insert, that will make the game run slower for end users and be more difficult than necessary to program and maintain, both of which are undesirable.

Fit Criterion: The data should be stored in a way that is easy for the game program to access, and the table structure should be able to easily accept new data from .csv files.

Priority: High

Dependencies: R#1

Requirement #: 3 (Create Database Schema and Tables)  
Requirement Type: Performance

Description: Database schema and tables must have quick access to data (that is, they must have fast performance)

Rationale: Data should not take a long time to access.

Fit Criterion: If testers notice that data is taking a noticeable amount of time to access, investigate the cause of that to determine if it’s related to the database schema and tables themselves.

Priority: High

Dependencies: R#1, R#2

Requirement #: 4 (Create Database Schema and Tables)  
Requirement Type: Look and Feel

Description: The database schema and tables need to look orderly, but the presentation of the schema and tables is ultimately not that important.

Rationale: The database schema and tables will not be viewed by end users, so they need to look good only to the staff which will be maintaining them behind the scenes.

Fit Criterion: If I (the staff maintaining the schema and tables behind the scenes) think that the schema and tables look good, then this requirement can be considered a success.

Priority: Low

Dependencies: R#1, R#2, R#3

Requirement #: 5 (Create Database Schema and Tables)  
Requirement Type: Maintainability and Support

Description: The database schema and tables should be easy to maintain and support over the long run.

Rationale: This program may run for several years and may undergo several transfers of responsibility.

Fit Criterion: Over the long run and in the moment, we can reflect on whether the schema and tables are easy to maintain and support. There may be some hurdles introduced by the Oracle APEX software, but overall, it should not be my fault if we run into any maintainability issues.

Priority: High

Dependencies: R#1, R#2, R#3

Requirement #: 6 (Create & Run Web Scraper, then upload scraped data to database)  
Requirement Type: Functional

Description: The web scraper should be able to gather all the necessary data from the internet and compile that data into a .csv file which I can then upload to the database schema to insert the values. As such, the .csv file must be compatible with the database.

Rationale: This program may need to be run every few years for several years and may undergo several transfers of responsibility, plus if the web scraper does not work as designed then the whole project falls apart.

Fit Criterion: Check the formatting of the text in the .csv file while developing the web scraper to ensure compatibility with the database. Also, check the web scraper’s programming to make sure that it can scrape the correct data in a proper manner.

Priority: High

Dependencies: R#1

Requirement #: 7 (Create & Run Web Scraper, then upload scraped data to database)  
Requirement Type: Performance

Description: Web scraper should be reasonably fast and work properly each time it is run.

Rationale: If the web scraper is not fast and/or not reliable, it will be difficult to maintain over the long term, which violates R#6.

Fit Criterion: Trained admins should agree that the web scraper is reasonably fast and works properly each time it is run.

Priority: High

Dependencies: R#1, R#6

Requirement #: 8 (Create & Run Web Scraper, then upload scraped data to database)  
Requirement Type: Usability

Description: Backend software that should be usable to trained admin(s).

Rationale: This will only be used by trained admins, not end users.

Fit Criterion: The web scraper will be easy for trained admins to use if we can just set it and forget it each time we have to run it.

Priority: Medium

Dependencies: R#1, R#6, R#7

Requirement #: 9 (Create & Run Web Scraper, then upload scraped data to database)  
Requirement Type: Look and Feel

Description: Backend software that should be presentable to trained admin(s).

Rationale: This will only be used by trained admins, not end users.

Fit Criterion: The trained admin can understand at a glance how to navigate and use the software.

Priority: Medium

Dependencies: R#1, R#6

Requirement #: 10 (Create Game Application)  
Requirement Type: Functional

Description: The game application should not only be a re-creation of the original budget game, but should also improve the game where it’s most needed like with visuals, for example.

Rationale: This project will be a replacement for the old budget game, so it needs to do all the same things that the old budget game did, but do them better than the old version.

Fit Criterion: Compare new budget game to the old one to see whether it is a faithful re-creation along with whether what needed to be improved was indeed improved upon.

Suggested Improvements:

1. Consolidate the entire game into one application without any extra discrete parts.

2. Integrate coin-flipping for the game into the software.

3. Integrate the necessary math calculations that user needs to perform into the software.

4. Integrate the user interactivity elements (such as choosing a career, etc.) and interface into the software.

Priority: High

Dependencies: R#1

Requirement #: 11 (Create Game Application)  
Requirement Type: Look and Feel

Description: The game app should be visually appealing and feel good to use for end users.

Rationale: The game app is the part of this project that will actually be used by end users, so the game’s visuals must be good in their eyes.

Fit Criterion: I can have people play the game app for testing purposes and during the testing, I can have them rate how much they like or dislike the visuals.

Priority: High

Dependencies: R#1, R#10

Requirement #: 12 (Create Game Application)  
Requirement Type: Performance

Description: The game app should perform well, that is, it should not be slow to run on a wide variety of devices.

Rationale: If the game app runs slowly, users will not have a fun time playing the game. Even if a user runs the game on a slow device, the game app should still run at a playable speed.

Fit Criterion: During the testing phase, I can have users with a variety of different devices test the game app and report how well they perceived the performance to be.

Priority: High

Dependencies: R#1, R#10, R#11

Requirement #: 13 (Create Game Application)  
Requirement Type: Performance

Description: The game app should also be built well in the sense that the code should be robust and fault-tolerant. The game’s code should not crash easily, but ideally not crash at all.

Rationale: If the game crashes while running and/or is not fault-tolerant, then it will cause a bad experience for the user and be problematic to try to fix on the fly.

Fit Criterion: During the testing phase, I can test for bugs as well as have users test for bugs. Based on the results of these tests, I can fix the code as needed to produce a more stable result.

Priority: High

Dependencies: R#1, R#10

Requirement #: 14 (Test project thoroughly as per test plan document)  
Requirement Type: Functional

Description: Any and all tests I run on this project should be good tests that return valuable information.

Rationale: Tests that do not return valuable information are a waste of time to run.

Fit Criterion: Any test that is run should return at least some information that contains value, which means that the information can be used to help improve the project.

Priority: High

Dependencies: All requirements in this document

Requirement #: 15 (Deliver Finished Product)  
Requirement Type: Functional

Description: The finished product should be delivered in a way that is easy for end users to use and is easy for me to maintain.

Rationale: If the finished product is not easy to use and/or is difficult to maintain, then that creates unnecessary problems and headache for all parties involved, which should be avoided.

Fit Criterion: Run tests with end users to get their opinions on whether the game is easy to use and navigate, and check up with admins every so often to get their opinions on whether the project is easy to maintain.

Priority: High

Dependencies: All requirements in this document

Requirement #: 16 (Deliver Finished Product)  
Requirement Type: Security

Description: Again, we need to make sure that the project doesn’t store any user data, as described in R#1.

Rationale: We’re checking for this again here in order to limit scope creep.

Fit Criterion: Make sure that all data stored is obtained from publicly-available sources and that there is no personal user information in the project database.

Priority: High

Dependencies: R#1, and the project needs to be finished by this point.