Cost Estimation for the Student Grade Management System using the COCOMO Model

a) Software Development Mode Type: Semidetached. The Student Grade Management system will need to manage data for the EKU's Registrar Office and allow users to perform functions such as adding, modifying, deleting, and printing student grade records. Additionally, the system will need to be able to import Excel files provided by faculty. These kinds of functionalities lean towards a System Development Mode of the Semidetached type. It is not as robust as the embedded mode type nor as rigid as the organic mode type. Some features will be tailored to the university's registrar office, but most of the features are common functions related to managing and importing data.

b) EAF Rating(s) or *Cost Drivers*:

Product Attributes:

1. RELY (Required Software Reliability): High (1.15) - Administration requires accurate data for calculating GPA.
2. DATA (Database Size): Nominal (1.00) - The database should only require three tables with several thousand records each.
3. CPLX (Product Complexity): High (1.15) - The product is tailored to the administrative needs of EKU's Registrar Office.

Computer Attributes:

4. TIME (Execution Time Constraint): High (1.11) - The system needs to respond in an acceptable amount of time.
5. STOR (Main Storage Constraint): High (1.06) - The system will require access to a database if not its own.
6. VIRT (Virtual Machine Volatility): Nominal (1.00) - The system requires no more than the average allocated resources of a typical data management system.
7. TURN (Computer Turnaround Time): Very High (1.15) - The developers will need responsive machines for development.

Personal Attributes: One person will wear several hats and perform each of these roles.

8. ACAP (Analyst Capability): Very High (0.71) - The analyst involved requires the aptitude to complete the project alone in less than 8 weeks (about 2 months) by himself.
9. AEXP (Applications Experience): High (0.91) - The developer needs to be experienced with the IDE being used to develop the system by himself.
10. PCAP (Programmer Capability): Very High (0.70) - The programmer needs to be experienced enough to integrate a database with a software application by himself.

11. VEXP (Virtual Machine Experience): High (0.90) - The network admin needs to be able to remotely access and program a database or tables of an existing database for the system by himself.
12. LEXP (Programming Language Experience): High (0.95) - The developer needs to be experienced with C# to design the system in less than 8 weeks (about 2 months) by himself.

Project Attributes:

13. MODP (Modern Programming Practices): High (0.91) - The product needs to adopt the latest practices in the industry.
14. TOOL (Use of Software Tools): High (0.91) - The product will be heavily reliant on the software tools available.
15. SCED (Required Development Schedule): High (1.10) - The project will need to be completed in less than 8 weeks (about 2 months) by one person wearing many hats.

c) Intermediate COCOMO Effort and Schedule Equation(s): Use Semidetached Development Mode Type Equations from the Intermediate COCOMO Model.
   a. Assume KDSI=300 (300,000 lines of code) as per the Semidetached slide from the Cost Estimation PowerPoint: *"The size range of a semidetached mode product generally extends up to 300 KDSI".* The choice is to consider the maximum cost of the project:
      i. Nominal Effort Value: 3.0*(300)^1.12=
      ii. Product of the EAF Multipliers: EAF=(1.15*1.00*1.15*1.11*1.06*1.00*1.15*0.71*0.91*0.70*0.90*0.95*0.91*0.91*1.04)=0.6303 or approximately 0.63
      iii. EAF=0.63
   b. Use EAF=0.60 and KDSI=300 in the below equations:
      i. Effort Equation: E=EAF*3.0*(KDSI)^1.12=(0.63)*3.0*(300)^1.12=0.63*3.0*594.81=1124.18
      ii. Schedule Equation: TDEV=2.5*(E)^0.35=2.5*(1124.18)^0.35=2.5*11.69=29.22
   c. The estimated effort and schedule for this project is roughly 1124 person-months and 29 months, respectively. These numbers appear to be exceptionally high. As an aside, how could one student complete this in 8 weeks? I believe each of values are large due to the chosen KDSI, but assuming that the Student Grade Management System were for an actual university and not just a class project, this would make sense. However, 300,000 lines of distributed code seems a bit excessive for a class project. Let's assume some lower values for KDSI:
      i. If we reduced the KDSI to say 100 (100,000 lines of code), the estimated effort and schedule for this project is reduced to roughly 328 person-months and 19 months (about 1 and a half years), respectively.
      ii. If we reduced the KDSI to say 50 (50,000 lines of code), the estimated effort and schedule for this project is reduced to roughly 151 person-months and 14 months, respectively.

iii. If we reduced the KDSI to say 10 (10,000 lines of code), the estimated effort and schedule for this project is reduced to roughly 25 person-months and 8-9 months, respectively.
iv. If we reduced the KDSI to say 5 (5,000 lines of code), the estimated effort and schedule for this project is reduced to roughly 11 person-months and 7-8 months, respectively.
v. If we reduced the KDSI to say 1 (1,000 lines of code), the estimated effort and schedule for this project is reduced to roughly (1.89) or 2 person-months and 3 months, respectively. This or less seems more appropriate for a single student developing a class project.