Dylan Cornelius, David Koger, James Travis

CSC 440

7 September 2017

Assignment 1

1. Tasks
   1. Deciding whether to take on project 1 or project 2: feasibility study to determine which project to take on. (1)
   2. Determine tasks: list out every task that is necessary for creating a system that satisfies the project’s needs. (2)
   3. Pert chart: Shows the paths to each task and milestone to complete the system. (1)
   4. Gantt chart: Shows the available timelines for each task which lead to the completion of each milestone. (1)
   5. Create scheduling report: Put together a scheduling report to show the customer the tasks and scheduling for the project. (1)
   6. Submit scheduling report: Show the report to the customer and receive feedback. (1)
   7. Review scheduling report: Use input on the scheduling report to make changes to the report. (1)
   8. Gather project information: Talk to customer to gain specific information about the system. (1)
   9. Write functional requirements: Specify all functional requirements of the system. (3)
   10. Write nonfunctional requirements: Specify all nonfunctional requirements of the system. (1)
   11. Use case diagram: determining the complete business transactions. (1)
       * Input:
       * Output:
   12. Doman diagram: shows classes, and their attributes, to show the customer the larger picture of the system. (1)
   13. Create requirements specification report: Put together a requirements specification report to show the customer for feedback. (1)
   14. Submit requirements specification report: Show the report to the customer and receive feedback. (1)
   15. Review requirements specification report: Use input on the requirements specification report to make changes to the report. (1)
   16. Sequence diagram: demonstrates the individual functions of the system. (3)
   17. State diagram: proves the status of the system during each step of the system’s functionality. (2)
   18. Class diagram: further enhances the domain diagram by showing the functions of each class. (1)
   19. Activity diagram: Shows the individual processes of each function. (3)
   20. Entity Relationship diagram: Explains to relationships between each database table. (2)
   21. Database schema design: Further specifies the database design from the ER diagram into a table format. (1)
   22. Data dictionary: Shows all pieces of data that will be utilized in the system. (2)
   23. Specify task list: Further pin point tasks after the Analysis Phase to more accurately design a schedule. (1)
   24. Designate tasks: According to the task list, assign specific people to work on specific tasks. (1)
   25. Research coding languages: Look at different coding languages to determine which one will work best for the system. (2)
   26. Decide coding languages: Determining what particular coding languages will be used to implement the system design based on research. (1)
   27. Design UI layout: Drawing out the system’s user interface and graphical design. (2)
   28. Create design specifications report: Put together the previous diagrams into a report to show the customer. (1)
   29. Submit design specifications report: Present the report to the customer and get criticism on it. (1)
   30. Review design specifications report: Change the report according to the customer’s feedback. (1)
   31. Build UI elements: Implement the user interface for the system. (5)
   32. Design UI elements: Graphically enhance the user interface of the system to make it more usable by the user. (5)
   33. Implement registration subsystem: Code the registration subsystem shown in the diagrams. (5)
   34. Implement file submission subsystem: Code the file submission subsystem shown in the diagrams. (3)
   35. Implement review
   36. Create database scripts: Implement the database design in the chosen database scripting language. (4)
   37. Connect database to methods: Add implementation of accessing the database to the functions in the system. (2)
   38. Basis path testing: Used to determine efficiency and unnecessary connections. (1)
   39. Evaluate cyclomatic complexity: Determining if the system is as efficient as it could be. (1)
   40. Condition testing: Testing each branch of the system to ensure it follows the desired path. (1)
   41. Data flow testing: Ensure that data moves properly between functions. (1)
   42. Equivalence testing: Tests every branch of data. (1)
   43. Boundary value analysis: Tests the endpoints of each branch of data. (1)
   44. Comparison testing: Tests that the same inputs give the same outputs. (1)
   45. Stress testing: Push the system to find out the system’s weak points. (2)
   46. Recovery testing: Testing the system to see how the system recovers from bugs. (1)
   47. Alpha Testing: Present the system to the customer and receive feedback. (1)
   48. Review project: Based on testing, make changes to system. (1)
   49. Beta Testing: Hand the system over to the customer to see what real-world bugs might still be in the system. (1)
2. PERT chart
3. TEs and TLs
4. Gantt chart