|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Comments** | **Mark (/100)** |
| 1448679 | Hanxiao Lin | Cover page, contents page, figure and tables numbered. Overall well presented. Quite a coarse grain design with 3 classes only. Little description of implementation, for example, pseudo-code describing the algorithms used. The testing description only comprises screenshots which are informative but a more systematic approach should be used including some brief information on computational efficiency. The greedy and basic ACO algorithm implemented. No use of inheritance to express the commonality between the algorithms. Some functionality demonstrated. A clearly laid out GUI. Wasn’t clear from your report whether you had implemented local optimization. Overall a good effort with a demanding assignment.  Use Case Model  Many of the requirements qualitative and not functional. No consideration of cost. Friendly convenient user interface is not a functional requirement. The user interface should also be convenient. No consideration of cost. Poor requirements analysis.  Poor Use Case View diagram. Operator override is not appropriate as a system response.  Splitting the user into observer and operator seems very artificial. The labels in the elipses do not correspond to appropriate system responses.  An excessive number of scenario descriptions. There should be at the most two. The scenario descriptions are passive and should be active. The scenario descriptions are not good.  I did not say that nouns might have too few attributes associated with them. This would be a property of classes. Selected classes are potentially ok but “Codes” is a difficult class to understand.  Aggregation is not an appropriate connection between classes in this case. A link would be ok.  CRC cards: I said do not use the phrase “”is responsible for” in the CRC responsibility description. The responsibility descriptions are vague and passive.  Sequence diagram: OK but the purpose of some messages is not clear.  The statechart contains several aspects of a flow chart. Drawing lines and points are not states. Algorithm and Exit response are not states.  The class diagrams of Figure 2.7.1 and 2.7.2 are over complex. Where have all these additional classes come from ?  Analysis Model  Reasonable set of attributes and methods. Do not see a need to manage the exit response.  Sequence diagram. It is not clear to me where the trigger events will arise.  Class diagram ok but aggregation is not necessary or appropriate. Having Execute with options as a method is clumsy. Would expect ExecuteGreedy() and ExecuteACO().  Statechart: Not convinced that these are states. Surprised by the many exit responses.  No non-functional requirements identified.  Design Model  Visibility in class diagram is ok. Ok not to develop sequence diagram. | 57 |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Comments** | **Mark (/100)** |
| 1487473 | Luis Enrique Ramos Maldonado | Cover page, contents page, figure and tables numbered. No page numbers. Appropriate introduction.  Overall well presented. A simple but adequate class selection and interaction. No pseudo code shown to illustrate algorithm design. You have shown useful screenshots to demonstrate functionality but a more systematic approach to testing is required where each component of your program is tested and documented. Only the greedy algorithm implemented. Quite well laid out code. Some functionality but inefficient implementation of greedy . It looks like the data has not been correctly pre-processed.  Good summary of initial requirements. Appropriate Use-case diagram.  Inappropriate to write CRC cards before selecting classes. I stated that the phrase “is responsible for” should not be used in the Responsibility description of a CRC card. DataFile is not an appropriate class or class name. The DataFile is an attribute.  Reasonable statechart. The actions and events leading to some transitions are questionable.  Either a collaboration or a sequence diagram would suffice. Sequence diagram with two superimposed state regions is not appropriate. Each state box should be on a separate branch of the time line.  Several message directions and names are not appropriate on the collaboration diagram.  On the sequence diagram returns to the user should be implied. A message to send the data is not needed. Data flow is normally left implied on a collaboration and sequence diagram. The sequence diagram message “Display Results” is sent to the user but the user cannot display the results.  Analysis Model  Comments as above.  Design Model  The sequence diagram notation: Greedy()/ACO()/ RefTable() is not consistent with UML notation.  All attributes should be private. The link notation of aggregation with nan arrow is not consistent with UML notation. I realise that you have used a design tool but I warned you against design tools that did not follow UML notation. | 56 |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Comments** | **Mark (/100)** |
| 1379396 | Pei Shi | Cover page but no contents page. Figures numbered correctly. Use of scanned diagram with no reference. Formatting could be better. Well written. Pages numbered. Margins excessively large. A lot of regurgitation of the assignment document describing the algorithm. 3 simple classes chosen. The use of ‘system’ for a class name is not appropriate. This is effectively the solver class. No algorithm implementation described nor pseudo-code given. Minimal testing which just comprises a screen shot with no information about algorithm parameters or the accuracy of the ‘solution’. The greedy algorithm has been efficiently implemented. ACO has also been implemented and seems to give sensible results on small datasets although the progress bar on the GUI is misleading.  Use Case Model  Requirements criteria not well presented. Priority is the interpretation of the other three and would naturally be at the end of the list.  Windows is a slightly unusual first system response. Having the system as an external entity is not appropriate.  Survey descriptions are well written.  Classes should be identified before CRC cards are written and the interaction diagram drawn.  I expressly stated do not use the phrase “is responsible for” in the CRC card responsibility description. The responsibility descriptions also lack clarity in places.  Sequence diagram is in part ok at this stage. Stating there is a do loop is not informative. The phrasing is strange.  Statechart: “Controller” is not a state. Start and end state’s not shown correctly. The transition labels are not appropriate; they do not identify the reason or action associated with the transition. They are too functional in nature.  Now you provide a class diagram with boxes for actions. This is not UML notation and it is far from clear how these classes were identified. The arrows are not appropriately used to link classes.  Analysis Model  Identification of attributes is not explained.  In the sequence diagram the trigger events are not appropriate.  Statechart diagram  The various actions in each state are not appropriately ongoing activities.  This is not clearly a state chart.  Comments on non-functional requirements identifies some issues.  Design Model  A fair set of attributes and methods.  Sequence diagram transition labels lack clarity. There is no sense of trigger event and action.  State chart not developed but improvement needed. Then a new state chart is given. The transitions are still mainly inappropriate and the chart is still not clearly a statechart.  Limited comment on the realisation of non-functional requirements. | 53 |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Comments** | **Mark (/100)** |
| 1481476 | Guanshen Yan | Cover page, contents page, figure numbers. Use of a scanned in figure with no reference. A Menu is an inappropriate name for what is a GUI (Form) class containing a menu control. No algorithm implementation described nor pseudo-code given. Your testing does have some systematic display screenshots but you should tabulate your results. Extensive code submitted with almost no commenting. You have put everything under a single Forms class so not really an object oriented solution. Efficient implementation of Greedy and also ACO has been implemented.  Use Case Model  No prioritisation of requirements.  Reasonable Use Case Diagram.  Scenario descriptions are passive and do not describe the users actions nor the system responses.  Default action is not a user focus for a use case scenario description.  Class diagram ok but aggregation is not an appropriate relationship.  I explicitly said do not use the phrase “is responsible for” in the CRC responsibility description. The description does not describe the actions of the current class and the response of other classes. New class introduced without an explanation.  Sequence diagram: Not clear that the “Solver” class is appropriately named but the messages are mainly appropriate. Appropriate use of implied returns.  The statechart is part state diagram and largely a flow chart. Display information is not appropriate as a state.  Analysis Model  Attributes and methods ok. Run algorithm would incorporate the subsequent messages.  Use of aggregation is not appropriate. Methods are not in the same place or the same as the previous and subsequent sequence diagrams.  The revised statechart is a little better. Trigger events are not clearly triggers and multiple trigger events in a chain are not appropriate.  Design Model  Non-functional requirements were not identified in the Analysis Model section. Those mentioned here are functional requirements. | 57 |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Comments** | **Mark (/100)** |
| 1429155 | Fan Jingqiao | Cover page, contents page, figure numbers. No algorithm implementation described nor pseudo-code given. Your testing does have some systematic display screenshots but you should tabulate your results. Extensive code submitted but all under a single Forms class which is not an object oriented solution. No inheritance implemented which would enable you to exploit the commonality between some methods and attributes in the solvers. Some functionality presented. A nicely laid out GUI. Greedy has been efficiently implemented and ACO has also been implemented although no indication about the accuracy of it’s final ‘result’.  Use Case Model  Use case view diagram communicates no sense of the user requirements. There is no actor and the labels on each arc are not appropriate nor are the descriptions of each node.  The survey description is vague, passive and indirect.  No class identification.  The classes in the collaboration diagram are not appropriate and are over connected. The standard notation is not used.  Statechart: These are not states. This is a very detailed and incomplete program flow description.  Activity diagrams are not needed. This adds nothing to the design.  CRC cards: You avoid the pitfall of stating what the class is responsible for. However the description is still passive and indirect.  Analysis Model  The method and attribute description does not define data types.  Sequence diagram: the message labels are far too long and describe issues that should be defined through separate messages. Selecting a starting point at random is not the same as randomizing the cities.  The linkage between classes is excessive and the directed links are not appropriate.  The statechart bears no resemblance to a state chart. All attributes should be private and structured data types are seldom appropriate. I note that the directed broken arrow has been dropped.  There is little state activity in this application.  Design Model  Attribute and method voisibi0lity not shown. Nob-functional requirements are not considered. | 47 |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Comments** | **Mark (/100)** |
|  | Shanthini Muthusamy | Cover page, contents page and page numbering. No ID number given anywhere in the document. Cover is incomplete. One equation appeared to be scanned into the report. This is inappropriate and unnecessary. Spaces used rather than tabbing.  Good selection of classes. Very well presented report. Pseudo code presented for the ACO algorithm. Thorough approach to testing with screenshots and tabulated performance data. A rather awkward and over complex GUI which I couldn’t get to run with the ACO algorithm. A simple menu system suffices. The Greedy algorithm selection seemed to run OK. Also no way of using the standard datasets as the data is selected via xml files.  Use Case Model  Requirements prioritisation largely ok. I would give a higher priority to the graphical display of city location.  Use Case View Diagram: I presume that by trail you mean trail. Several system response are not appropriate. I do not understand how “visit all cities” can exist alongside “chose next city” and “start city”. Develop GUI is not a response of the operating program. No indication as to how solver is selected and how data is acquired.  Scenario description: I would expect one scenario description for the one user. The listed points of reference do not fit the design process. The scenario description needs to be more sharply focussed. This leads to the excessive list of potential classes. Final set of classes is mainly ok but the ant class might be excessive.  I am not convinced by the responsibility descriptions. Having classes calculate the distance to neighbouring classes will result in double the number of calculations necessary. The responsibility description are passive not direct and active.  The collaboration diagram is ok but the above comments regarding the city class apply. Don’t need collaboration and sequence diagram.  Statechart is more a flow chart than a statechart.  Analysis Model  Class diagram: the XML file is not a class nor a specialisation of a GUI class. Links rather than aggregation would suffice. The structure is ok but over complex. Reasonable attributes and methods but earlier comments still apply.  Attributes and methods should be identified before they are used.  Reasonable development of collaboration and sequence diagram. As before one or other needed. Some classes not appropriate.  Statechart: Suitably developed but not appropriate as a statechart.  Explanation of how associations are identified does not depend on attributes.  No need for packages.  Design Model  Revised class diagram difficult to read. Visibility is mainly ok. Not appropriate for all methods in a class to be private. Strange to have a class with one or more methods and no attributes and no methods and no attributes.  Final sequence diagram should show triggers. | 65 |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Comments** | **Mark (/100)** |
| 1487897 | Luciano Rivera | Cover page. No contents page or page numbering. Figures numbered. Top level headings not numbered.  Very well presented report overall. Class selection lacks a class representing the city grid (problem domain). Filereading is not an appropriate class name. Some implementational details given but use pseudo code rather than code snippets to illustrate algorithm design. Your test description clearly shows screenshots which are informative but some tabulated performance data would be useful . The solver class comprises methods for each algorithm rather than using inheritance which is more object oriented. Some functionality. The GUI only seems to display the final result and not update the route as the algorithm runs. At least a progress bar should be added to make the interface more user friendly.  Review of requirements is reasonable. No prioritisation of requirements.  Use Case Model  Use case vie diagram is very simple, perhaps too simple  Good scenario description.  Good class identification.  Class diagram is ok. Aggregation is not really necessary but is ok.  I explicitly said do not write “is responsible for” in the CRCD card responsibility description. You have done so and as a result the descriptions are passive.  Either a collaboration or a sequence diagram would suffice. Message to user should be implied.  Reasonable statechart.  Analysis Model  Attributes and methods not in the CRC card responsibilities.  Class diagram and visibility is good.  Outline of methods is noted. | 68 |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Comments** | **Mark (/100)** |
| 1458881 | Dongling Yang | Cover page but no page numbering. Figures numbered. Poor composition. English spelling is Modelling not modeling. The first four pages are redundant. The first sentence of the second paragraph is a tautology.  The report contains no explanation of implementation, for example pseudocode showing algorithm design. Testing is not systematic as it is basically just a set of screenshots. Rather an overcomplex class layout. The inheritance relationships seem inappropriate. Solver classes should be independent of the Form class as they could potentially be console applications. A very elaborate GUI. Efficient implementation of Greedy. Also ACO seems to be working correctly .  Use Case Model  The structure of the program is described before it has been designed.  Use case diagram is simplistic. The scenario description presumes the design and does not start with requirements.  The Use case class diagram is basic but ok.  CRC cards. I expressly said never write “is responsible for”. This is a passive construct and is not appropriate.  Statechart is more a flow diagram.  Sequence diagram is ok.  Analysis Model  Attributes and methods: Too much in the TSP Window class. The design should have been extended to more classes.  Class diagram aggregation using “kind of algorithms is not appropriate. More appropriate to be a link.  Statechart is still more a flow diagram.  The sequence diagram is ok. But could have been developed further.  Design Model  No attributes should be global. | 61 |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Comments** | **Mark (/100)** |
| 1486148 | Matteo de Cicco | Cover and contents page, page numbering but no figure or table numbering. Correct class layout with inheritance relationship linking different solvers. Very good use of pseudocode to indicate algorithm implementation. Excellent structured testing including screenshots. Some good functionality displayed with a clear GUI and implementation of different ACO algorithms and Greedy although there doesn’t seem to be much difference in the final route length! It would be nice to see route update as the algorithm progresses rather than at the end of processing.  Use Case Model  Simple review of requirements and no prioritisation.  Use case view diagram is very simple, over simple.  Scenario description should be active but is passive. Classes should be identified at this stage but are not.  CRC cards: I stipulated that you should not write “is responsible for” in the responsibility description. You have done so and as a result the description passive and indirect. This hinders attribute and method identification. Classes not identified but classes used for CRC cards.  The belated class diagram does not show all the classes. Composition is not necessary. Now the class diagram is expanded. There are far too many classes.  Statechart: Most of the boxes are states. This design does not have significant state aspects. Transitions are appropriate.  Sequence diagram: The actors are not appropriate; they are not users. The messages are ok.  Analysis Model  Back to split class diagram. Aggregation is not necessary. Inheritance is appropriate. Could use template classes.  Attributes and methods are mainly ok. The notation for get or set versions of several methods is not appropriate.  Statechart: Do activity is not appropriate.  Sequence diagram: Use of condition notation for iteration is not appropriate. Choice of actors is different again and still not appropriate.  Design Model  Visibility of attributes and methods is ok. | 69 |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Comments** | **Mark (/100)** |
| 1483063 | Wang Jing | Cover page, no contents page. Page and figure numbering. Section 4.3 is blank. Rather a long introduction. What’s an international diagram? No description of algorithm design using pseudocode. You have attempted some structured testing but it is useful to tabulate results as well as including screenshots. Good use of inheritance in linking solver classes. Simple clear GUI. Efficient implementation of Greedy and looks like ACO is performing correctly also.  Use Case Model  Use case view diagram is good. Very detailed. Actor not labelled.  Scenario description: The various roles or actors are not appropriate. “Operator override is far from appropriate. Not in a direct style.  Class ID of program is not appropriate. Reduction to two classes is severe.  I explicitly said do not write “is responsible for” in the CRC card responsibility description. The descriptions are passive and indirect and therefore not fit for purpose.  Sequence diagram: too few classes but otherwise the messages are mainly ok. There is no need for a message to transfer data. Data flows regardless of the direction of the message.  Sate chart: These are not states.  Analysis Model  Attributes and methods not from CRC cards because the responsibilities were not appropriate.  Sequence diagram: comments as above.  Class diagram: Aggregation is not appropriate.  Statechart: comments as above.  Design Model  Visibility of attributes and methods is ok. | 60 |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Comments** | **Mark (/100)** |
| 1337236 | Yechen Tian | Cover page. No contents page but page and figure numbering. Superficial class diagram on page 10. Not sure what this is telling me. The report contains no explanation of implementation, for example pseudocode showing algorithm design. Testing is not systematic as it is basically just a set of screenshots. You could have tabulated your programs performance in terms of accuracy and speed. Your program is basically a single GUI class. Not really an OOP solution. Efficient implemention of Greedy. Nice simple GUI with constant route update. Your ACO solver seems to get stuck in an infinite loop and doesn’t reduce the route length.  Strong similarities to 1379396. Whilst different words are used the structure and presentation be. Whilst different words are used the structure and presentation bears string similarities. This will be reviewed. Any repetition of this degree of similarity in the future would be a serious matter.  Layout could be better; Several Figure captions come before the diagram on a separate page. Margins very large. Cover page and page numbers but no contents page.  Use Case Model  Outline of classes from nouns and stereotype considerations. No final justification of selected classes.  I expressly said do not use the phrase “is responsible for” in the class responsibility description. The descriptions are passive.  Sequence diagram is basic but ok for the use case model.  The satechart could be better but is a reasonable start for a Use Case Model.  Class diagram is satisfactory but should have been shown much sooner.  Analysis Model  Sequence diagram event/action notation does not follow UML notation. It is largely satisfactory.  Class diagram is ok but the notation for attributes is not clear.  Statechart notation is non-standard. Start state not shown correctly. No end state. It is not clear what is intended by some actions and trigger events.  Design Model  Should be more methods and attributes in the class diagram. These classes should be linked by association. Aggregation is not appropriate. | 52 |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Comments** | **Mark (/100)** |
| 1417886 | Yang Song | No cover page or contents page. Page and figure numbering. English spelling is Modelling not modeling.  Poor composition. A lengthy and unnecessary overview of UML given. The report is a description of ACO with no substantive design or implementation described and nothing on testing. Your program is basically a single GUI class. Not really an OOP solution. Nice simple GUI with constant route update. Your ACO solver seems to get stuck in an infinite loop and doesn’t reduce the route length.  The 6 plus pages of general introduction to software design, programming and UML were not needed. Discussion of Cobol, Yourdon and Ward methods is redundant. Programming languages might be devised but they are not born. Smallpox is not a programing language. Perhaps you meant “Smalltalk”. I have no idea what you are claiming that I said about crushed state diagrams.  There is a vague description over 2.5 pages of what is to be designed. This could and should have been about 1 side.  The assessment of requirements was vague. There is no explanation of why the risk was high the key requirements were not clearly identified, nor their cost nor importance. There is no explanation of why the risk was high the key requirements were not clearly identified, nor their cost nor importance.  Use case diagram inappropriate.  No explanation of class choice. The choices are inappropriate.  Sequence diagram is ok for the chosen classes. I am puzzled by the action to chose an algorithm.  State chart is more a flow chart.  No distinction of Use Case Modelling, Analysis Modelling and Design Modelling. Many steps o-mitted;} No CRC cards, no development of class diagram, sequence diagram and state chart. No identification of methods or attributes. | 39 |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Comments** | **Mark (/100)** |
| 1460340 | James Nkhata | Cover, contents page. Page numbering. Well presented overall. Some use of pseudocode for algorithm description although lacking in detail. Testing description is mainly screen shots with no tabulated performance data. You have a single solver class with different algorithm being different methods of Solver. This is a procedural solution. Nicely laid out code which is well commented. 2 day late penalty. A very elaborate GUI. Doesn’t look like it’s displaying the route correctly. I couldn’t verify functionality as the program hung after a short time.  Use case design  Good assessment of requirements priority.  Use Case Model  There are too many actors in the use case view diagram. Algorithm applier / solver is not appropriate. The scenario descriptions should be active and direct but are passive.  Nouns selection: The passive style leads to inappropriate nouns that are not appropriate as classes.  This is not the point at which to identify attributes. Need CRC card responsibilities first.  CRC cards: I explicitly said do not write “is responsible for” in the CRC card responsibility. I also said that the responsibilities should be direct and active., They are not.  Statechart: most of the states are not valid states, e.g. Problem Display, Solution Display, Problem Data Loaded and Solution Generation.  Sequence Diagram  Classes changed with no explanation. You need to be more systematic. This is unacceptable. The labelling of separate phases is not necessary.  Analysis Model  Reasonable attributes and methods.  Sequence diagram: There is no reason to number the messages they should be in order. This was very clearly explained in the lectures. I get the impression that you must not have been present or not listening. I do not understand the purpose of an implied message to the controller class.  Class diagram is ok.  Statechart: Instantaneous actions should not be labelled as “Do:”. Still not a good statechart.  Non-functional requirements as listed are functional.  Design Model  Visibility of attributes. All should be private. Having DistanceList as public is inappropriate.  This report demonstrates a poor understanding of the design process. | 50 -10=40 |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Comments** | **Mark (/100)** |
| 1505175 | Jolyon Joe Ng | A well written report including a cover page, contents list can correctly labelled figures and tables. Your design seems a nit fine grained. Coordinates should be attributes of a class representing the problem domain (the City grid) and not a separate class. Good use of pseudo code and some evaluation testing has been included based mainly on program screenshots. A correct separation of the solution into the 3 projects. Also good use of inheritance in your Solver classes. Your GUI is nice and simple but my main issue is your functionality. You have not allowed the use of standard TSP datasets to adequately test your code. The 3 datasets you provide are rather small and they show no difference between the greedy algorithm and the ACO algorithm in performance. | 62 |