**Assignment Feedback 2012**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Comments** | **Mark (/100)** |
| 0939057 | S. Bishnoi | Report: No cover sheet. Contents page but no main section numbering. Excellent use of pseudo-code to explain algorithms. Use of invoke and embedded resources. The use of a generic for the queue is not necessary but using a queue container for the lift requests is sensible. Detailed explanation of test results are not backed up with corresponding outputs (such as screen dumps). Test figures should not be in the appendix. No substantive conclusions have been made.  Functionality: Nice GUI. The lift seems to only respond fully to floor buttons and not to lift buttons. The simulator doesn’t do anything. The animation is not smooth.  Design  No cover sheet.  Use-case: The Use-case scenarios are good but could include additions for a key holder to override and an emergency event. I would not have chosen the specialise option but you have used it appropriately. In the second set of diagrams the structure is different but I cannot find an explanation for the change  Scenario Description: It would be better to have one scenario description per use-case. That is not critical in most cases but what is important is that the scenario description needs to be a “dialogue” between the system and user; The user does x and the system responds by doing y, with the various conditions and explanations included. The fragmentation of the scenario descriptions means that you have created special cases when they are not strictly necessary. This is likely to com0plicate the later design outcomes.  Class selection: I thought you had a good set of classes well justified from the emboldened items plus 1 but then I found the CRC cards went on. I think what you have is ok because several of the classes are concerned with the simulation but it is not clear how you identified these. That is because the use-case considered only the lift and not the simulation. The collaboration diagram gives a better understanding of the classes. A class diagram would have helped in this case. Having classes for the lift and Landing buttons is ok but not essential. Having a button class and no lift or floor display seems asymmetric.  CRC Cards: The responsibilities are ok but could be better. I did say avoid using “Responsibility” in the text as this leads to a passive style when a direct active style is required. The style here is a little passive and not focussed on the how the class is required to respond in each case. The final class diagram of Figure 15 is good and the visibility of attributes and methods is good. The use of inheritance is misplaced.  Interaction Diagrams: The initial collaboration diagram is fine. The messages should be numbered in sequence. Figure 12 does not show important returns for door closed. Actor is not a package. I guess this is a feature of the design tool that you used. Figure 14: a collaboration diagram needs messages, there are none.  State Chart: Not bad. Has some elements of a flow chart. An important state not shown is lift idle. I don’t think that joins are necessary; there is no joining of threads. Activity diagrams are rarely necessary and I think are used here to express what could be ain an interaction diagram.  Attributes and Methods: The methods and attributes should come from the CRC cards. The reason this was difficult was the style in which the responsibilities were written. That having been said it may be that you think better with pictures than lists or text. You have a good set of attributes and methods.  Simulator designed. | 65 |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Comments** | **Mark (/100)** |
| 962001 | S. Crooks | Report: Very well presented with a cover sheet, contents page and section/sub-section labelling. Excellent use of pseudo-code. A good description of methodical testing for each module but not backed up with example screen dumps or console output. Well laid out and documented code with correct use of event handling and multithreading. No evidence of result of testing but a good set of tests with detailed observation.  Functionality: Seems to work perfectly including the event generator. Simple GUI but contains all the relevant interactions and information. Smooth animation.  Design  Use-case: Well done for considering the simulator in the use-case. Event generator response, start and stop plus set interval I expect to be part of the simulations use-case, which you have restricted to the event generator. Three of these responses appear in both use-cases. This is not automatically a problem but I cannot see why it is appropriate here. The GUI does not seem appropriate as a user  Scenario Description: In normal use the user would not necessarily see the direction that the lift is travelling until the lift has been called. That is normally a concern for the controller. The scenario descriptions need to be more direct, active and detailed in style then identifying classes would be easier.  Class selection: I think you needed a display for the simulator. This was not identified because the scenario description lacked critical detail.  CRC Cards: The responsibility descriptions need to be active, direct and more detailed.  Class diagram: This should show the connections between classes as one diagram.  Interaction Diagrams: These diagrams seriously lack detail.  State Chart: Lift moving to selected floor or floor from which it was called is really the same state of lift moving (to destination). Figure 6: the buttons represent the event generator stopped. The state chart could be better  Simulator designed. | 70 |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Comments** | **Mark (/100)** |
| 1015440 | L. Peel | Report: Cover page but no contents page or section numbering. Some good use of pseudo-code to describe implementation. The addition LiftCar class seems redundant – should be part of the LiftController. Use of threading and event handling OK. Testing is adequate with use of screenshots.  Functionality: Simple but clear GUI. Full functionality including simulator. Not sure the purpose of the 0 and 1 matrix in the top right of the GUI. No additional controls to change simulator mean event time or lift speed. Smooth animation.  Design  **Use-case:** It would be good to have a main system response in each scenario. Pre and post conditions are not commonly a part of UML. They draw the designer into premature decisions.  **Scenario Description:** Not generally good practice to use the specification to identify classes.  Class selection: You would have been well advised to look for stereotypical classes. In a sense this is what you have done without identifying it as such.  **CRC Cards:** None. This is a critical gap. Attributes and methods that were added later were appropriate as too was their visibility.  Interaction Diagrams: I cannot track each use-case in the sequence diagram. The recurrent use of Fire event is not helpful.  **State Chart:** None.  Simulator not designed. | 59 |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Comments** | **Mark (/100)** |
| 104316 | Jihoon Han | Report: Fair presentation with a cover sheet, contents page and section/sub-section labelling. Looks like the program is essentially a GUI with no lift controller or simulator projects. Most of the lift control is done in the Control\_Manager method. The testing description includes supporting screen shots. Nice use of pseudo-code to describe the lift scheduling algorithms but rather many programming constructs in the pseudo code. Testing with simple expected inputs.  Functionality: Simple GUI. Fully functional program except for a runtime crash when the simulator is paused. Smooth animation. The GUI needs to display more information about mean event time and lift speed. The GUI form doesn’t close down gracefully as it looks like the threads have not been aborted on window closure.  Design  **Use-case:** The use-case is strong in this case. Therefore, resort to domain and business models is less important. Notation used for include is that for generalisation. Stereotypical links should be denoted by double chevrons, ≪stereotypical link≫. The sub-responses for the button select do not relate to the higher level responses. The roles are not well defined. It is unconventional to then revise the use-case diagrams. The new diagrams are not an improvement. The “valve” symbols are not part of standard UML notation. The additional roles are not appropriate.  **Scenario Description:** Text descriptions are not needed in addition to scenario descriptions. The textual descriptions are too vague and brief to help with the design process. The Scenario descriptions on add a little and are not direct and active in style. There are more scenario descriptions than use-cases. This is strange as there should be a 1:1 correspondence.  **Class selection:** mainly ok but the process by which you arrived at these classes is not convincing. It is mainly down to your judgements rather than following the process.  **CRC Cards:** Reasonable but I cannot follow the operation of the lift by a user. Attributes could be better named. Not buttons but some attribute of the button would be a class attribute. I would expect the simulator attributes to be separated from the lift attributes.  **Interaction Diagrams:** The messages on a collaboration diagram should be numbered in order. Do not need both collaboration and sequence diagrams. Loops back to the start are not required. Use conditions instead.  **State Chart:** The state chart states match the classes in the collaboration diagram. This is not appropriate.  Notes: Final class definition, attributes and methods are satisfactory. Use of aggregation is not appropriate. Later diagrams seen but not needed in this case. | 62 |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Comments** | **Mark (/100)** |
| 1052698 | Y. Gu | Report: Very well presented with cover page, contents and section numbering. Excellent use of pseudo-code to describe the lift scheduling algorithm. Nicely coded with full use of event handling and threading although there is a lack of commenting in the code. The description of testing needs to be supported with screenshots. No testing of unexpected input. Limited testing with expected input.  Functionality. Nice GUI. Full functionality including the simulator. Use of synthesised speech is a useful addition. Additional controls for lift speed and mean event time.  Design  The poor grammar makes parts of the report difficult to understand. Composition. I suspect that this is more about allowing time for proof reading than anything else given that elsewhere the report is clear.  Use-case: Lacks detail. Role of each actor should be specified. The use-case for passenger in lift and on the landing lack detail.  Scenario Description: The poor use-case selection and representation leads to a poor focusing of the scenario descriptions.  Class selection: States is not a good class. No simulator classes.  CRC Cards: What you have written as a description should be in the Responsibility box and should be better focussed on a set of specific behaviours and not generalisations.  Class Diagram: The class diagram should show class links using standard links where possible. Aggregation could be used to good effect. Poor attribute and method selection. Class diagram of Figure 8 is missing many methods and attributes. Visibility is ok. No relationships between classes shown.  Interaction Diagrams: The messages do not seem appropriate in Figure 4. I do not expect the display to update the controller or buttons to return an illuminate message back to the controller. I do not expect the controller to send a reach floor message to the lift. Do not need both collaboration and sequence diagrams. The passenger class should be an actor.  State Chart: The state chart of Figures 6 & 7 is trivial.  Simulator not designed. | 68 |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Comments** | **Mark (/100)** |
| 1135155 | N. Viknaswaran | Report: Cover page, contents page, section numbering. Figure captions. Figure 2.1 seems to combine a use case and class diagram. Seems a complex implementation with a thread for each floor button which is unnecessary. The testing description describes tests that were carried out but not the test results. Clearly the bugs in the lift operation have not been revealed by the testing. Some appropriate use of pseudo-code to describe the lift scheduling. The conclusion has no summary of project outcomes.  Functionality: Incomplete functionality. The lift stops responding to requests after the first few have been handled. Also the lift can disappear out of the window. Not clear the purpose of the extra 2 buttons. Lift direction information displayed on the GUI.  Design  The introduction lacks foocus.  **Use-case:** Reasonable use case except Display and Sensor are not appropriate use-cases.  **Scenario Description:** Presumes the design a little. Not quite written as a user-system response dialogue.  **Class selection:** Initial list ok. Reasons for rejection could be better articulated. Final class list ok but SenseDirection is not appropriate as a class. It is a method. Class diagram uses novel notation. This is not satisfactory.  **CRC Cards:** Responsibilities should be in an active and direst style not passive.  **Interaction Diagrams:** In the collaboration diagram the messages carry too many steps in the title. Do not need sequence and collaboration diagram. The late introduction of the sequence diagram is inappropriate.  **State Chart:** Nested states are not necessary. Aspects of notation are unconventional.  Simulator not designed. | 53 |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Comments** | **Mark (/100)** |
| 1148183 | O. Folorunsho, | Report: Cover page, contents page but no section numbering. No figure captions. It seems that much of the report is a technical summary of the classes/methods and could have been automatically generated. These are better placed in an appendix. No need to include program listing. The design is over complex. Avoid classes as with state only and no functionality. Rather lengthy code with some empty ‘if’ clauses! Good use of inheritance for the button sensors. Testing should be supported with program output examples. No substantive conclusions.  Functionality: Nice GUI with clear buttons. Doesn’t seem to be fully working. The lift doesn’t visit floor 3. The lift speed doesn’t change in response to the button. The order of visiting doesn’t seem correct. Smooth animation. Clear icons. The simulator seems to work but again, the mean event time doesn’t change.  Design  **Use-case:** Needs more detail but a reasonable start.  **Scenario Description:** Not well tied to use-case diagrams. Does not focus sufficiently on user and system responses.  **Class selection:** The two lists are ok but it is not clear what classes were chosen and why. 6 classes is the upper limit of what is appropriate  **CRC Cards:** These come late in the process. Simplistic responsibility descriptions are not adequate to identify attributes and methods.. No collaborators for the sensor class means that it is not connected to another class.  **Interaction Diagrams:** Messages in collaboration diagram should be numbered. Do not need collaboration and sequence diagrams.  **State Chart:** More of a partial flow chart.  **Class Diagram:** Some classes not connected. Attribute and method visibility is largely ok.  Simulator not designed. | 57 |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Comments** | **Mark (/100)** |
| 1166500 | Y. Dai | Report: Cover page but no contents page. Section numbering. He has combined the design document with his programming report. Some pseudo code presented but it is not clear. Seems like the code is mainly a GUI with no separate controller class as specified. No genuine description of testing although lots of bugs reported. The use of code snippets in the main text is not an acceptable way of presenting a report.  Functionality: GUI shows doors opening/closing but no lift movement animation. Includes synthesised speech. Seems to respond correctly to floor and lift button requests. Lift panel indicators working correctly. Simulator included. Responds correctly to lift speed button.  Design  The business and domain models are not covered by a class diagram.  **Use-case:** The use –case diagram is very confused and adds little to the design process.  **Scenario Description:**  None  **Class selection:** Not described.  **CRC Cards:** None  **Interaction Diagrams:** The sequence diagram is trivial.  **Class Diagram:** It is strange to have a class diagram before classes have been selected and without any relationships between classes.  **State Chart:** None.  Simulator not designed. | 45 |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Comments** | **Mark (/100)** |
| 1210686 | I. ObazeleI | Report: : Cover page, no contents page, section numbering. Figure captions. Rather brief description of the lift controller class and no description of the gui class. I couldn’t find the ‘Transportation’ class as referred to in the document. Some very awkward coding with multiple ‘if’ statements which could have been done better. Use of event and threads appropriate. A very thorough description of testing with associated screen shots. The tabulated results are well presented. A good summary of the achievements is presented in the conclusions.  Functionality: Only partial functionality. The lift doesn’t seem to respond correctly to lift button requests. Seems to handle floor button requests OK. Also it stops responding at some point. There doesn’t seem to be a simulator implemented. Quite a basic GUI with lift motion indicator panel.  Design  **Use-case:**  Lift arrives destination and no pending requests are not good use-cases. Otherwise the diagram is good.  **Scenario description:** Could be better written as a “dialogue” between the user and the system. <<Lift car>> is not a suitable stereotype. Not clear how the number of classes was reduced.  **CRC cards:** An active and direct style of responsibility description is required.  **Interaction diagrams:**  Unusual to show the actor with states. Not necessary to have both a sequence and collaboration diagram. DoorClose() as a message to the controller seems dis directed; better to LiftCar. Jump of control from 5 to R between Control and LiftCar and back to Control between 7 and 8. This seems to happen in the Collaboration and sequence diagram.  **State Chart:** More a flow chart.  **Class diagram:** Too many public attributes. Some methods are private that I think need to be public, eg Check LiftStatus()  Simulator not designed. | 67 |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Comments** | **Mark (/100)** |
| 1225316 | C. Han | Report: Cover page but no contents page and no section numbering. Some attempt at describing lift scheduling algorithms but not proper pseudo-code. No discussion of class relationships or even what classes have been created. Some code commented out but otherwise, inadequate commenting. Rather awkward coding with long if sequences. Uses threads and events appropriately. Reasonable account of testing .  Functionality: Full functionality. Nice clear GUI. Smooth animation. Responds to lift speed button. Not exactly sure what the other button does. Simulator included. Lift panel and floor indicators give useful information.  Reasonable account of testing in the design section  Design  **Use case diagram:** Good choice of scenarios. You might have included an external event such as a fire alarm and a key override. I presume “Red” should be “Read”. The use of extends should be include.  **Scenario descriptions:** are not labelled as such and written in a passive voice when they should be in an active and direct style. They address appropriate issues. Good arguments for class selection except a “state” is not a class.  **CRC cards:** Responsibilities address appropriate issues but again are written in a passive voice and should be written in an active direct style.  **Attributes:** Most are named as though they are methods.  **Methods:**  Ok but no queue management.  **Interaction diagrams:** A sensor would not normally cause a button state to be set, possibly a display. Button state class is poorly named. There is a jump of control from A0 to A1. Strange to get state after just setting state. No B0 or C0 but there is an A0 and two C1’s. No need for a collaboration diagram and a sequence diagram.  Simulator not designed. | 70 |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Comments** | **Mark (/100)** |
| 1228236 | Y. Liang | Report: Cover page and contents page and section numbering. The report lacks detail about the classes that have been designed and how the algorithms were implemented. Good description of algorithm implementation using pseudo-code. Threading and event handling implemented. The test description has some screenshots but there are no formal test results presented and the account of testing seems superficial to me.  Functionality: Full functionality. Nice clear GUI with additional fire alarm button. Smooth animation. Simulator working. Clear lift and floor button indicator panels. The lift doesn’t respond to the change of lift speed setting on the slider.  Design  **Use-case:** Fire alarm is an external event not a system response. No indication of how the lift is moved.  **Scenario Description:** None.  **Class selection:** Not justified at any point.  **CRC Cards:** Should be used to select attributes. The responsibility descriptions lack detail and are difficult to understand due to poor composition.  **Interaction Diagrams:** Do not need sequence and collaboration diagrams. Neither show how lift movement is managed. Messages in the sequence diagram should be in order on the time line and therefore do not need numbering.  **Class diagram:** shows no class connections and lacks detail.  **State Chart:** Figure 2 is far too early in the design process.  Simulator not designed. By definition there should not be two states with the same name. This chart is confused and lacks detail. | 62 |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Comments** | **Mark (/100)** |
| 1230666 | Y. Feng | Report: Cover page. No contents page. No section numbering. The flow chart of figure 1 conveys little useful information and a flow chart is not a good way to present an object-oriented program. The figure captions are incomplete. Most of the report is concerned with the GUI layout rather than a detailed description of implementation. No discussion about the classes. Some pseudo-code presented. Rather simple code in the lift controller project. Looks like most of the scheduling is done by the GUI. Not thoroughly tested and no formal presentation of test results although some screenshots (which don’t tie up with observed functionality) have been included.  Functionality: Only partially working. Seems the lift simulator works but the lift doesn’t respond to floor or lift buttons. The GUI is well laid out and there is smooth animation. Clear indicator panels on the lift and floor buttons.  Design  No contents page. No page numbers. Sections not numbered. Table headings should be above the Table.  **Use-case:** Basic but ok.  **Scenario Description:** lack detail and passive.  **Class diagram:** Initial class diagram is ok. Not clear how the classes were selected. In Figure 6 additional classes have been added with no justification.  **Class selection:** Poor explanation of class selection. Minimal set of classes. Simulator not considered.  **CRC Cards:** Responsibility descriptions lack focus and detail.  Interaction Diagrams: Messages in a collaboration diagram should be numbered. There is no need for a collaboration and sequence diagram.  **State Chart:** Trivial.  There is no need for an activity diagram. This is a simplistic flow chart that adds nothing to the design process.  Simulator designed as a late addition | 51 |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Comments** | **Mark (/100)** |
| 1243156 | G. Xu | Report: Cover page, contents page but no section numbering. No figure captions. Sensible use of ArrayList to handle button requests. No clear summary of the main classes in the application. Good use of threading and event handling. No extensive testing described although the scenarios chosen are sensible. Some screenshots illustrating correct execution would be useful. No need to include code.  Functionality: Very clear GUI. Full functionality. Lift speed and mean event time changeable from the GUI. Smooth animation. Simulator included. Good use of icons and colour.  Design  **Use-case:** Could basic scenarios. Could have added external event and key override.  **Scenario descriptions:** Good. Reducing the design to one class was a very poor decision.  **CRC cards:** Now you write CRC cards for two classes. This is puzzling. The responsibilities are not appropriately written in a direct active voice as complete statements. The attributes and methods are all plucked from thin air.  **Interaction diagrams:** To simple due to too few classes. Do not need both collaboration and sequence diagram.  **State chart:** states could be better named and chosen but not too bad. Transitions should have more annotation.  Simulator not designed. | 71 |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Comments** | **Mark (/100)** |
| 1280872 | A. Al-Ghafri, | Report: Cover page and contents page and section numbering. The report lacks detail about the classes that have been designed and how the algorithms were implemented. There is a small amount of pseudo-code but it needs more detail and substance. Rather simple lift controller code which suggests the GUI is doing the work. The GUI should just reflect state of the lift and no more. A reasonable effort on testing with use of screenshots and various scenarios tested.  Functionality: Full functionality. Clear GUI layout with useful lift and button status indicators. Simulator working. Smooth animation. Lift speed and mean event time buttons implemented correctly.  Design  **Use-case diagram:** There is no basis for having use case responses detached from any external entity. Move stop does not relate to Check display as extends. Likewise it does not relate to Update list of calls by extends. Is the text at the foot of page 2 a scenario description. It is not identified. It is written passively which is not appropriate.  **Scenario description:** It is not appropriate move the lift immediately a button has been pressed. In the use case you showed a list of calls but this concept has not been included here where it most belongs.  **State chart:** Again the transition to the lift to move is inappropriate. Simulator, Lift and Button event are not states but entities, perhaps classes.  **CRC cards:** The Responsibilities are not written in a way that helps identify methods and attributes. The collaborators of Lift controller are inconsistent with the class diagram. The collaborators of Main Form are not consistent with the class diagram. There is no button event class. This name indicates that this is not an appropriate concept for a class. There is no Lift state in the class diagram. This by definition is not an appropriate concept for a class.  Simulator not designed. | 61 |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Comments** | **Mark (/100)** |
| 1286893 | S. Mavlenov | Report: Cover page, contents page, section numbering. Figure captions. Rather lacking in detail concerning the structure of the main classes which make up the application. Lots of pseudo-code presented. Use of threading and event handling. Overall, efficiently coded. Some screenshots have accompanied the test description but test results could have been better presented. Speed cannot be pixels.  Functionality: Simple GUI. No floor labels. Full functionality. Lift button speed can be set. Mean event time input textbox awkward. Increment and decrement button easier for the user. Useful lift state information shown on GUI.  Design  **Use-case:** Very good use-case diagrams.  **Scenario Description:** Should be a direct and active description rather than passive. The lift controller should also be queuing and directing the lift movement. They should be written for each use case. Instead you seem to have pre-empted the identification of tasks.  **Class selection:** Good class selection and justification.  **CRC Cards:** The responsibility description needs to be direct and active in style. “Responsible” has steered the descriptions to be passive. Move up and Move down and several other items seem more likely to be methods than attributes  **Interaction Diagrams:** Messages on an interaction diagram should be numbered in sequence. Do not need both sequence and collaboration diagrams. Reach floor is likely to be a real message or a status setting.  **State Chart:** Good but I do not consider Lift Up and Lift Down to be needed as states. This can be covered by a parameter being set. “No Request” is a parameter value; an empty list, not a message.  **Class Diagram:** I presume that the class relationships are uses. Aggregation would be appropriate in some cases. Methods should be public in most cases  Simulator not designed. | 74 |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Comments** | **Mark (/100)** |
| 1287212 | R. Song | Report: Cover page. No contents page. Section numbering. Pseudo code to show how lift scheduling algorithm operates although it is not necessary to explain it at each floor! No discussion of class relationships or even what classes have been created. Seems to be awkwardly coded. Not clear why so many different floor events need to be handled. Also the code is almost completely devoid of comments. Account of testing in design report is out of place. The testing is superficial.  Functionality: Full functionality. Clear GUI although no floor labels. Smooth animation. All lift buttons operational. Simulator working. Lift speed button select works. No way to alter event time. Includes useful emergency button. Why should a sensor supply button information ?  Design  **Use-case:** This should come before the scenario descriptions. Superficial.  **Scenario Description:** First line is redundant. The scenario descriptions are confused and lack detail.  **Class selection:** Classes identified do not derive from the scenario description and are not appropriate. Stereotypical classes more appropriate. I do not understand the reason for a “Driver” class. I am surprised that there is no lift class. Some of the other classes are questionable as lacking serious content. No class for the simulator.  **CRC Cards:** Responsibilities are difficult to understand and lack directness. They are passive when they should be active in style. It is not clear how the attributes and methods were derived from the CRC card responsibility descriptions.  **Interaction Diagrams:** Collaboration diagram has jumps in control from message A2 to A3, A6 to A7, B3 to B4, D1 to D2, D5 to D6. Do not need both collaboration and sequence diagram. Message pattern should not be different for collaboration and sequence diagrams.  **State Chart:** This is more a flow chart, of sorts.  Class diagram: Attribute and method visibility is not clear. Notation is not UML notation.  Simulator not designed.  Test account in design report is out of place. The account of testing is sparse. | 58 |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Comments** | **Mark (/100)** |
| 1301931 | M. Ahmed Marsuik | Report: Section heading indentation is unusual. Cover page and contents page but no section numbering. Some figure captions missing. Code snippets in main text rather than pseudo-code. No discussion of class relationships or even what classes have been created. The code is poorly presented with commented out segments. Seems like appropriate use of events and threads. The discussion of testing is more like a discussion about how the GUI was developed. No conclusions have been discussed.  Functionality: No simulator. Nice graphics. No floor numbering. No indicators showing lift status and floor/lift buttons don’t show when they have been pressed. Smooth animation.  Design  **Use-case:** The long chain is not quite correct. Move to next floor is not a sub-response of door close. FloorSensor and DirectionSensor is not appropriate as an actor.  **Scenario Description:** Well written  **Class selection:**  Good initial reasoning. Two classes is too little.  **CRC Cards:** Good responsibilities. Reasonable attributes and methods.  **Interaction Diagrams:** Oversimplified due to just two classes. Does not use the object-oriented paradigm to good effect. Do not need both collaboration and sequence diagram. Visibility of attributes and methods is good.  **State Chart:**  States not well selected. The state chart is more of a flow chart. The join would is not necessary as part of the state chart.  Simulator not designed. | 54 |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Comments** | **Mark (/100)** |
| 1238163 | R. Fang | Report: Cover page and contents page and section numbering. Some good use of pseudo-code to illustrate algorithm implementation. Not very much detail on object oriented aspects not classes used. A lot of the report is really a user manual. Appropriate use of threading and event handling. The test description is minimal with some screenshots but no formal test results presented.  Functionality: Full functionality. Clear GUI, smooth animation. Useful lift and floor button indicators. The lift speed button seems to have no effect. Simulator working correctly. No pause/play or event time buttons added to control the simulator.  Design  The domain model has too much technical content. Why is there a book mark in the scenario description?  **Use-case:** Excellent diagrams  **Scenario Description:** The scenario description need more work to detail the user requests and response.  **Class selection:** No explanation.  **CRC Cards:** Responsibility description should be direct and active and more detailed, not passive. Too little detail to identify attributes and methods.  **Interaction Diagrams:** Do not need sequence and collaboration diagram. The sequence diagram is good  **State Chart:** Ok but too early in the document.  **Class diagram:** Having the Floor and count attributes public is difficult to justify. More methods need to be public. No public methods in Buttons and Display classes means that they cannot receive a message.  Simulator not designed. Booch design book in reference list is not appropriate for UML design. | 62 |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Comments** | **Mark (/100)** |
| 1307339 | S. Pooni | Report: Well presented with cover page, contents and section numbering. No need to add code in the appendix. Use of xaml to create GUI, which hadn’t been taught. Clear description of program implementation. Informal use of pseudocode. Good use of event handling and threading. The lift control is split over several classes. The testing descriptions should be supported with screenshots. No substantive conclusions presented.  Functionality. Very nice GUI with good use of icons for clarity. Full functionality. Use of additional buttons to control mean event time and lift speed.  Design  **Use-case:** Lift sensor and floor sensor are not appropriate actors. They all lack a great deal of detail.  **Scenario Description:** Direct and active but should describe more of the process rather than have separate scenario descriptions for moving the lift etc. This is a detail and at least this aspect is described.  **Class selection:** A sound approach with many good decisions. I think you needed Controller, GUI and Lift classes. I note that the final list includes classes that were ruled out in Table 2. Consistency is needed. Floor sensor and Queue are questionable as classes. Good revision of Classes in Section 4.  **CRC Cards:** The responsibility description is detailed direct and detailed in style. Good selection of methods and attributes.  **Interaction Diagrams:** Sequence diagrams are good. The loop box is not necessary. Loops are effected by the conditions given to messages. I do not understand the selectLift loop in Figure 4. Figures 5& 6 make it clear that the use case for floor and lift sensors was a mistake.  **State Chart:** I do not believe that a nested state chart is necessary. There is a strong suggestion that this is wrong from the outer state label “Do something on GUI”. Good state chart for controller class. I think the processing of adding destinations could be in the one Adding requests state. This will ultimately simplify the design. In practice the “lift Class” could be part of the controller but there are reasons to separate it out. I do not believe that a nested state is need here. The “Do something interesting” state does not change the design.  The class links could be aggregation from GUI to Floor sensor and to Lift  Simulator designed. Visibility of methods and attributes is good. | 74 |