**INFORMATION TECHNOLOGY**

**DATA VALIDATION TASK MARK SHEET**

|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | **Description** | **Possible**  **Mark** | **Actual**  **Mark** |
| **Interface Design Requirements (GUI)**  Screen is clearly laid out - font, colour, font size, button to perform validation (3)  Fields aligned and clearly labelled (2).  Screen has a title and centered on the screen (1).  Each input component has an associate label (2) | Use the four categories to allocate marks.  -1 for each error. | **8** |  |
| **Variety of Types Used**  Four Data Types on GUI - String, Numeric, Boolean and Date/Time. | Deductions to a maximum of 4  (-1 for each type of missing field) | **4** |  |
| **Choice of appropriate components for each data type**  Numeric, String, Boolean, Date/Time | Deduction to a maximum of 4  (-1 for each missing component) | **4** |  |
| **Valid reason for choice of each component**  Numeric, String, Boolean, Date/Time | Deduction to a maximum of 4  (-1 for each missing/incorrect reason) | **4** |  |
| **Naming Conventions**  Labels, buttons, fields all correctly named according to conventions. | Deductions to a maximum of 2  (-1 for each incorrectly named field) | **2** |  |
| **Programming code with possible exception handling for each validation rule**  Correct, working code for each of the four validation rules. (4x2)  Rules achieve what is described in the reason for the rule (4x1). | 3 marks for each rule.  -1 to a max of 3 for each error in each rule. | **12** |  |
| **Descriptive error message for each rule**  Error message is provided with a detailed explanation. (4x1).  Each error message is placed in a label next to the incorrect component. (2)  If all input is valid, a confirming message is displayed (2) | Deduction to a maximum of 2 per incorrect error message.  Deduct 1 mark for a vague message per rule. | **8** |  |
| **Testing for each rule**  Choice of standard, extreme and abnormal data to test each rule (1x4)  Evidence of testing (1x4) | Deduct 1 mark for incorrect data per rule  Deduct 1 mark for lack of evidence per rule | **8** |  |
| **TOTAL** | | **50** |  |

**INFORMATION TECHNOLOGY**

**DATA VALIDATION TASK**

The first row includes an example of an age field. Please do not use this example, as this value can be calculated from a date.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name of Field and Type** | **Input Component** | **Reason for Input Component** | **Data Validation – Name of Rule (e.g. Format, length,**  **presence, check digit, range, type, logic)** | **Reason for Data Validation** |
| Example: age : integer  (Do NOT use this example) | Text field | User can enter any age | Range Check | Age must lie between 0 and 125 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**IFORMATION TECHNOLOGY**

**DATA VALIDATION TASK**

The first row includes an example of an age field. Please do not use this example, as this value can be calculated from a date.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name of Field and Type** | **Input Component** | **Reason for Input Component** | **Data Validation – Name of Rule (e.g. Format, length,**  **presence, check digit, range, type, logic)** | **Reason for Data Validation** |
| Movie date: date | JXDatePicker | Is user friendly and user can pick any date without having to type in the date directly.  Prevents the user from entering data that is not a valid date. | Presence Check  Range check | The user must select a date (Cannot be left empty)  Movie date chosen cannot be a date before todays date |
| Number of seats: Integer (numerical) | JSpinner | User can choose the number of seats without typing in anything.  Prevents entering of invalid data.  The number of seats can also be limited using this component. | Range check | Cannot book more than 20 seats at a time. |
| 3D glasses: Yes or No (Boolean) | JRadioButton grouped by a ButtonGroup | Only a Yes or No option is provided.  User can only click one of the buttons.  Prevents user from entering data that is not a yes or a no | Presence check | User has to select either yes or no and cannot leave this option blank |
| Phone Number: String | JTextField | A phone number starts with 0 and therefore cannot be an Integer but a string. User can type in any phone number | Presence check  Length check  Format check | This field cannot be left blank  Must consist of 10 numbers  First digit must be a zero and all digits must be numbers |

|  |  |  |  |
| --- | --- | --- | --- |
| **Name of Field** | **Standard** | **Extreme** | **Abnormal** |
| Movie Date | Todays date | In a months time | Past date |
| Number of seats | 2 | 20 | 25 |
| 3D glasses | yes | no | Don’t click anything or try click both |
| Phone number | 0825467232 | 0000000000 | Doesn’t start with 0, incorrect length or left blank. Contains text |