**Task Document 1**

*Amendments to Tables*

1. In the ClientContactTable, after AggregateRating, please add a new field:
   1. NumberOfRatings, Integer.
2. In the ProviderContactTable, after AggregateRating, please add a new field:
   1. NumberOfRatings, Integer.
3. In the ProviderContactTable, after NumberofRatings, please add a new field:
   1. ProviderOrganisation, Integer.
4. In the ProviderOrganisationTable, please add the following column after “Email2”
   1. Website, String.
5. In the ServicesProvidedTable, please add the following column after “Service”
   1. Suburb, String.
6. Create a new table “DiscussionTable”:
   1. DiscussionID, Integer (unique identifier, primary ID).
   2. EntryDate, DateTime.
   3. SenderID, String.
   4. RecipientID, String.
   5. Message, String.

*Amendments to Login component*

1. General description:
   1. Providers are made up of the organisations (which might be a company) and the actual people (who are listed in the contacts).
   2. When I gave the initial instructions, I said that the ‘login’ need to check with the ProviderContactTable – ie each individual user can log into the system
   3. Individual ProviderContacts will represent people who can be linked to a ProviderOrganisation, and removed from it again, as individuals who are ‘in the system’ move from ProviderOrganisation to ProviderOrganisation.
   4. When a ProviderContact logs in, we want the “Provider.php” page they see to take information from the ProviderOrganisation they are associated with.

*Amendments to Page*

1. These changes need to be made to each of the following files:
   1. Client.php
   2. Provider.php
   3. Agency.php
2. Each page should be set out roughly as follows:

|  |  |  |
| --- | --- | --- |
| Logo | ManageCare | Loggedinname  Logout Button |
| Menu | Operative Area | |

Where:

* 1. Place a circle in the area marked Logo at present – this is a placeholder.
  2. Place the words “ManageCare” in the middle of the top of the page at present.
  3. In the top righthand corner
     1. Loggedinname is the FirstName and LastName of the person who is logged in, taken from the relevant table, ie:
        1. ContactContactTable;
        2. ProviderContactTable; or
        3. AgencyOfficerTable.
     2. Place the logout button in the top righthand corner.
  4. All menu options need to go into a list placed in the menu area.
  5. All information displayed as a result of choosing a menu options while logged in needs to go to the Operative Area.

*Client.php: Amendments*

1. In Client.php, the items in the menu area are:
   1. Account Details
   2. Current Plan
   3. Current Bookings
   4. Search
   5. Discussion

*Provider.php: Amendments*

1. In Provider.php, the items in the menu area are:
   1. Account Details
   2. Current Bookings
   3. Services
   4. Discussion

*Agency.php: Amendments*

1. In Agency.php, the items in the menu area are:
   1. New Client
   2. Review Discussion

*Client.php menu time: Account Details*

1. In Client.php, when someone clicks “Account Details”:
   1. Highlight the Account Details menu item
   2. Display the following information in the Operative Area
2. Within the Operative Area, display:
   1. Row 1:
      1. Label, named NDISNumberLabel, with text: “NDIS Number”
      2. Label, named NDISNumberValue, with text: ClientContactTable.ClientNDISNumber
   2. Row 2:
      1. Label, named NameLabel, with text: “Name:”
      2. Textbox, named: FirstNameTextBox, with default text: ClientContactTable.FirstName
      3. Textbox, named: MiddleNameTextBox, with default text: ClientContactTable.MiddleName
      4. Textbox, named: LastNameTextBox, with default text: ClientContactTable.LastName
   3. Row 3:
      1. Label, named BirthDateLabel, with text: “Date of Birth:”
      2. Textbox, named: BirthDay, with default text: the day component of ClientContactTable.BirthDate
      3. Textbox, named: BirthMonth, with default text: the month component of ClientContactTable.BirthDate
      4. Textbox, named: BirthYear, with default text: the year component of ClientContactTable.BirthDate,
   4. Row 4:
      1. Label, named AddressLabel, with text: “Address:”
      2. Textbox, named: UnitNumberTextBox, with default text: ClientContactTable.UnitNumber
      3. Textbox, named: StreetNumberTextBox, with default text: ClientContactTable.StreetNumber
      4. Textbox, named: StreetNameTextBox, with default text: ClientContactTable.StreetName
      5. Textbox, named: StreetTypeTextBox, with default text: ClientContactTable.StreetType
   5. Row 5:
      1. Textbox, named: SuburbTextBox, with default text: ClientContactTable.Suburb,
      2. Dropbox, named: StateDropBox, with options ACT, NSW, NT, QLD, SA, TAS, VIC, WA, and default chosen option is the value of ClientContactTable.State
      3. Textbox, named: PostcodeTextBox, with default text: ClientContactTable.Postcode
   6. Row 6:
      1. Label, named Email1Label, with text: “Email 1:”
      2. Textbox, named: Email1TextBox, with default text: ClientContactTable.Email1
   7. Row 7:
      1. Label, named Email2Label, with text: “Email 2:”
      2. Textbox, named: Email2TextBox, with default text: ClientContactTable.Email2
   8. Row 8:
      1. Label, named MobilePhoneLabel, with text: “Mobile Phone:”
      2. Textbox, named: MobilePhoneTextBox, with default text: ClientContactTable.MobilePhone
   9. Row 9:
      1. Label, named HomePhoneLabel, with text: “Home Phone:”
      2. Textbox, named: HomePhoneTextBox, with default text: ClientContactTable.HomePhone
   10. Row 10:
       1. Label, named PreferredContactLabel, with text: “Preferred Contact Method:”
       2. Textbox, named: PreferredContactTextBox, with default text: ClientContactTable.PreferredContact
   11. Row 11:
       1. Label, named SecurityQuestionLabel, with text: “Security Question:”
       2. Textbox, named: SecurityQuestionTextBox, with default text:ClientContactTable.SecurityQuestion
   12. Row 12:
       1. Label, named CommencementDateLabel, with text: “Joined ManageCare:”
       2. Label, named CommencementDateValue, with text: ClientContactTable.CommencementDate
   13. Row 13:
       1. Label, named AggregateRatingLabel, with text: “Aggregate Rating:”
       2. Label, named AggregateRatingLabelValue, with text: ClientContactTable.AggregateRating
   14. Row 14:
       1. Label, named Numberof RatingsLabel, with text: “Number of Ratings:”
       2. Label, named NumberOfRatingsLabelValue, with text: ClientContactTable.NumberOfRatings
   15. Row 15:
       1. Button, named UpdateDetailsButton, with caption “Update Details” and “on clicking”:
          1. Make a text box appear with:

* The caption “Are you sure you want to update your details?”
* Button “yes”
* Button “no”
  + - 1. If click “yes”, then call a function that updates each item in the ClientContactTable with the item on the page, ie:
* ClientContactTable.FirstName = FirstNameTextBox.value
* ClientContactTable.MiddleName = MiddleNameTextBox.value

… etc

1. If the user clicks on anything that changes what is displayed in the Operative Area, or logs out, or changes what is on the screen then:
   1. Check if any of the information in the boxes has been changed
   2. If yes, then produce the text box asking for acceptance of updates, with yes and no options
   3. If click “yes”, then call a function that updates each item in the ClientContactTable with the item on the page, ie:
      1. ClientContactTable.FirstName = FirstNameTextBox.value
      2. ClientContactTable.MiddleName = MiddleNameTextBox.value
      3. … etc

*Client.php menu time: Current Plan*

1. In Client.php, when someone clicks “Current Plan”:
   1. Highlight the Current Plan menu item
   2. Identify the ClientContactID of the person logged in, and search the NDISPlanTable for each row where:

NDISPlanTable.Client = ClientContactID.value; and

PlanEndDate.value = null

* 1. Display the following information in the Operative Area

1. Within the Operative Area, display for each row in the NDISPlanTable, a row with the following information:
   1. Row 1
      1. Label “Start Date”
      2. Label “Service”
      3. Label “Approved”
      4. Label “Available”
   2. Row 1+X:
      1. PlanStartDate
      2. Service
      3. AmountApproved
      4. AmountAvailable

*Client.php menu time: Current Bookings*

1. In Client.php, when someone clicks “Current Bookings”:
   1. Highlight the Current Bookings menu item
   2. Identify the ClientContactID of the person logged in, and search the BookingTable for each row where:

BookingTable.Client = ClientContactID.value; and

DateEnd.value = null

* 1. Display the following information in the Operative Area

1. Within the Operative Area, display for each row in the BookingTable, a row with the following information:
   1. Row 1
      1. Label “Provider”
      2. Label “Service”
      3. Label “Last Update”
      4. Label “Start Date”`
      5. Label “Price”
      6. Label “Frequency”
   2. Row 1+X:
      1. ProviderOrganisationTable.TradingName, for the row in the ProviderOrganisationTable where ProviderOrganisationID = ProviderContactTable.ProviderOrganisation.
      2. Service
      3. LastUpdated
      4. DateStart
      5. TransactionPrice
      6. If “Recurring” != null then Frequency

*Client.php menu time: Search*

1. In Client.php, when someone clicks “Search”:
   1. Highlight the Search menu item
   2. Display the following information in the Operative Area
2. Within the Operative Area, display the following items
   1. Row 1:
      1. Label, named: ServiceLabel, with text: “Service:”
      2. Dropbox, named: ServiceDropBox, where the options are identified as follows:
         1. Search each row of the ServicesProvidedTable.
         2. Include those rows where ServiceProvidedTable.Active = Yes.
         3. Exclude any duplicate entries in ServicesProvidedTable.Service
         4. Add the resulting list to the ServiceDropBox, with “Any” as the first entry
   2. Row 2:
      1. Label, named: SuburbLabel, with text: “Suburb:”
      2. Textbox, named: SuburbTextbox, which is blank
   3. Row 3:
      1. Label, named SuburbRangeLabel, with text: “Distance:”
      2. Dropbox, named: SuburbRangeDropbox, with values:
         1. Any
         2. 5
         3. 10
         4. 15
   4. Row 4:
      1. Button, named “SearchButton”, with text: “Search”, and on clicking, calls a search function.
3. On clicking on the SearchButton, search the ServicesProvidedTable and return any row where (noting that the search criteria are “and”, if any entry has been added to the search):
   1. ServicesProvidedTable.Service = ServiceDropBox.value
   2. ServicesProvidedTable.Suburb = SuburbTextbox.value
   3. Any suburb that is within the number of kilometres in SuburbRangeDropbox.value from the suburb named in SuburbTextbox.value.
   4. Display headings:
      1. Row 1
         1. Label “Logo”
         2. Label “Provider”
         3. Label “Phone”
         4. Label “Email”
         5. Label “Website”
         6. Label “Rating”
   5. For each returned row, display a row with the following information:
      1. Row 1+X: for the row where ProviderOrganisationTable.ProviderOrganisationID = ServicesProvidedTable.ProviderID:
         1. The image stored in the location contained in ProviderOrganisationTable.DisplayLogo
         2. The value in ProviderOrganisationTable.TradingName

* Note: the wording displaying the name needs to be presented as a hyperlink
  + - 1. The value in ProviderOrganisationTable.MobilePhone
* Or, if null, ProviderOrganisationTable.WorkPhone
  + - 1. The value in ProviderOrganisationTable.Email1
      2. A hyperlink to the website in ProviderOrganisationTable.Website
* Note: clicking on the website hyperlink should open a new page
  + - 1. The value in ProviderOrganisationTable.AggregateRating
  1. On clicking the hyperlinked name 🡪 moves to the Discussion section, with the provider. Store the following information
     1. ClientContactID
     2. ProviderContactID relevant to the one selected

*Client: Discussion Area*

1. Within the Operative Area, display the following.
2. On the righthand side, a scroll bar that allows the user to scroll to the bottom of the operative area.
3. The default position should always be the bottom of the scrolling area, so new messages are added at the bottom.
4. Search the DiscussionTable for each entry where:
   1. Sender = ClientContactID or ProviderContactID
   2. Recipient = ClientContactID or ProviderContactID
5. Added to the operative area, with the scroll window on the righthand side the following:
   1. Row 1: (in this case, counting from the bottom):
      1. A textbox, named MessageTextbox, into which the user can type messages
      2. Button, “Send”, on clicking, create a new row in DiscussionTable, such that:
         1. DiscussionID = next available integer.
         2. EntryDate, as it is.
         3. Sender = ClientContactID.
         4. Recipient = ProviderContactID.
         5. Message = MessageTextbox.value.
         6. Search the DiscussionTable for updating the messages in the operative area
   2. Row 2+ (in this case, counting from the bottom):
      1. Add each DiscussionTable.Message which was found on the search, in chronological order, ie the newest at the bottom.

*Provider: Account Details*

1. In Provider.php, when someone clicks “Account Details”:
   1. Highlight the Account Details menu item
   2. Display the following information in the Operative Area
2. Within the Operative Area, display:
   1. Row 1:
      1. Label, named NDISNumberLabel, with text: “NDIS Number”
      2. Label, named NDISNumberValue, with text: ProviderContactTable.ProviderNDISNumber
   2. Row 2:
      1. Label, named NameLabel, with text: “Name:”
      2. Textbox, named: FirstNameTextBox, with default text: ProviderContactTable.FirstName
      3. Textbox, named: MiddleNameTextBox, with default text: ProviderContactTable.MiddleName
      4. Textbox, named: LastNameTextBox, with default text: ProviderContactTable.LastName
   3. Row 4:
      1. Label, named AddressLabel, with text: “Address:”
      2. Textbox, named: UnitNumberTextBox, with default text: ProviderContactTable.UnitNumber
      3. Textbox, named: StreetNumberTextBox, with default text: ProviderContactTable.StreetNumber
      4. Textbox, named: StreetNameTextBox, with default text: ProviderContactTable.StreetName
      5. Textbox, named: StreetTypeTextBox, with default text: ProviderContactTable.StreetType
   4. Row 5:
      1. Textbox, named: SuburbTextBox, with default text: ProviderContactTable.Suburb,
      2. Dropbox, named: StateDropBox, with options ACT, NSW, NT, QLD, SA, TAS, VIC, WA, and default chosen option is the value of ProviderContactTable.State
      3. Textbox, named: PostcodeTextBox, with default text: ProviderContactTable.Postcode
   5. Row 6:
      1. Label, named Email1Label, with text: “Email 1:”
      2. Textbox, named: Email1TextBox, with default text: ProviderContactTable.Email1
   6. Row 7:
      1. Label, named Email2Label, with text: “Email 2:”
      2. Textbox, named: Email2TextBox, with default text: ProviderContactTable.Email2
   7. Row 8:
      1. Label, named MobilePhoneLabel, with text: “Mobile Phone:”
      2. Textbox, named: MobilePhoneTextBox, with default text: ProviderContactTable.MobilePhone
   8. Row 9:
      1. Label, named HomePhoneLabel, with text: “Home Phone:”
      2. Textbox, named: HomePhoneTextBox, with default text: ProviderContactTable.HomePhone
   9. Row 10:
      1. Label, named PreferredContactLabel, with text: “Preferred Contact Method:”
      2. Textbox, named: PreferredContactTextBox, with default text: ProviderContactTable.PreferredContact
   10. Row 11:
       1. Label, named SecurityQuestionLabel, with text: “Security Question:”
       2. Textbox, named: SecurityQuestionTextBox, with default text:ProviderContactTable.SecurityQuestion
   11. Row 12:
       1. Label, named CommencementDateLabel, with text: “Joined ManageCare:”
       2. Label, named CommencementDateValue, with text: ProviderContactTable.CommencementDate
   12. Row 13:
       1. Label, named AggregateRatingLabel, with text: “Aggregate Rating:”
       2. Label, named AggregateRatingLabelValue, with text: ProviderContactTable.AggregateRating
   13. Row 14:
       1. Label, named Numberof RatingsLabel, with text: “Number of Ratings:”
       2. Label, named NumberOfRatingsLabelValue, with text: ProviderContactTable.NumberOfRatings
   14. Row 15:
       1. Button, named UpdateDetailsButton, with caption “Update Details” and “on clicking”:
          1. Make a text box appear with:

* The caption “Are you sure you want to update your details?”
* Button “yes”
* Button “no”
  + - 1. If click “yes”, then call a function that updates each item in the ProviderContactTable with the item on the page, ie:
* ProviderContactTable.FirstName = FirstNameTextBox.value
* ProviderContactTable.MiddleName = MiddleNameTextBox.value

… etc

1. If the user clicks on anything that changes what is displayed in the Operative Area, or logs out, or changes what is on the screen then:
   1. Check if any of the information in the boxes has been changed
   2. If yes, then produce the text box asking for acceptance of updates, with yes and no options
   3. If click “yes”, then call a function that updates each item in the ProviderContactTable with the item on the page, ie:
      1. ProviderContactTable.FirstName = FirstNameTextBox.value
      2. ProviderContactTable.MiddleName = MiddleNameTextBox.value
      3. … etc

*Provider: Current Bookings*

1. In Provider.php, when someone clicks “Current Bookings”:
   1. Highlight the Current Bookings menu item
   2. Identify the ProviderContactID of the person logged in, and search the BookingTable for each row where:

BookingTable.Provider = ProviderContactID.value; and

DateEnd.value = null

* 1. Display the following information in the Operative Area

1. Within the Operative Area, display for each row in the BookingTable, a row with the following information:
   1. Row 1
      1. Label “Client”
      2. Label “Service”
      3. Label “Last Update”
      4. Label “Start Date”
      5. Label “Price”
      6. Label “Frequency”
   2. Row 1+X:
      1. ClientContact .
      2. Service
      3. LastUpdated
      4. DateStart
      5. TransactionPrice
      6. If “Recurring” != null then Frequency

*Provider.php: Amendments*

1. In Provider.php, the items in the menu area are:
   1. Account Details
   2. Current Bookings
   3. Services
   4. Discussion

*Provider: Discussion Area*

1. Within the Operative Area, display the following.
2. On the righthand side, a scroll bar that allows the user to scroll to the bottom of the operative area.
3. The default position should always be the bottom of the scrolling area, so new messages are added at the bottom.
4. Search the DiscussionTable for each entry where:
   1. Sender = ClientContactID or ProviderContactID
   2. Recipient = ClientContactID or ProviderContactID
5. Added to the operative area, with the scroll window on the righthand side the following:
   1. Row 1: (in this case, counting from the bottom):
      1. A textbox, named MessageTextbox, into which the user can type messages
      2. Button, “Send”, on clicking, create a new row in DiscussionTable, such that:
         1. DiscussionID = next available integer.
         2. EntryDate, as it is.
         3. Sender = ClientContactID.
         4. Recipient = ProviderContactID.
         5. Message = MessageTextbox.value.
         6. Search the DiscussionTable for updating the messages in the operative area
   2. Row 2+ (in this case, counting from the bottom):
      1. Add each DiscussionTable.Message which was found on the search, in chronological order, ie the newest at the bottom.

*Provider: Account Details*

1. In Provider.php, when someone clicks “Account Details”, create a new page entitled “Provider Account Details”, for the provider who has logged in with a given ProviderContactID:
   1. At the top of the page, please include a button for “Back”, which takes you back to the Provider.php page.
   2. Row 1:
      1. Label: “NDIS Number”
      2. Label: ProviderContactTable.ClientNDISNumber
   3. Row 2:
      1. Label: “Name:”
      2. Label: ProviderContactTable.FirstName, ProviderContactTable.MiddleName, ProviderContactTable.LastName
   4. Row 3:
      1. Label: “Address:”
      2. Label: ProviderContactTable.UnitNumber, ProviderContactTable.StreetNumber, ProviderContactTable.StreetName, ProviderContactTable.StreetType
   5. Row 4:
      1. Label: ProviderContactTable.Suburb, ProviderContactTable.State, ProviderContactTable.Postcode
   6. Row 5:
      1. Label: “Email 1:”
      2. Label: ProviderContactTable.Email1
   7. Row 6:
      1. Label: “Email 2:”
      2. Label: ProviderContactTable.Email2
   8. Row 7:
      1. Label: “Mobile Phone:”
      2. Label: ProviderContactTable.MobilePhone
   9. Row 8:
      1. Label: “Home Phone:”
      2. Label: ProviderContactTable.HomePhone
   10. Row 9:
       1. Label: “Preferred Contact Method:”
       2. Label: ProviderContactTable.PreferredContact
   11. Row 10:
       1. Label: “Security Question:”
       2. Label: ProviderContactTable.SecurityQuestion
   12. Row 11:
       1. Label: “Joined ManageCare:”
       2. Label: ProviderContactTable.CommencementDate
   13. Row 12:
       1. Label: “Aggregate Rating:”
       2. Label: ProviderContactTable.AggregateRating
   14. Row 13:
       1. Label: “Number of Ratings:”
       2. Label: ProviderContactTable.NumberOfRatings

*Provider: Services*

1. In Provider.php, when someone clicks “Services”, create a new page entitled “Services”, for the provider who has logged in with a given ProviderContactID:
   1. At the top of the page, please include a button for “Back”, which takes you back to the Provider.php page.
   2. Search the ServicesProvidedTable and for each row where ServicesProvidedTable = ProviderContactID, produce the following:
   3. Row:
      1. Label: “Service”
      2. Label: ServicesProvidedTable.Service

*Provider: List of Current Bookings*

1. In Provider.php, when someone clicks “Services”, create a new page entitled “Services”, for the provider who has logged in with a given ProviderContactID:
   1. At the top of the page, please include a button for “Back”, which takes you back to the Provider.php page.
   2. Search the BookingTable for each row where:
      1. BookingTable.Provider = ProviderContactID; **and**
      2. BookingTable.DateEnd > current date.
   3. Set up the following headings:
      1. Client
      2. Service
      3. Date
      4. Price
      5. Recurring
      6. Frequency
   4. For each row found in step (3) above:
      1. In the Client column, put in the clients’ name, which is found by:
         1. search the ClientContactTableID for the value stored in BookingTable.Client.
         2. For the relevant ClientContactTableID, insert the values for:

* ClientContactTable.FirstName
* ClientContactTable.LastName
  + 1. In the Service column: BookingTable.Service
    2. In the Date column: BookingTable.DateStart
    3. In the Price column: BookingTable.TransactionPrice
    4. In the Recurring column: Bookingtable.Recurring
    5. In the Frequency column: Bookingtable.Frequency