**Task Document 1**

*Amendments to Tables*

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

*Amendments to Login component*

* General description:
* Providers are made up of the organisations (which might be a company) and the actual people (who are listed in the contacts).
* 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
* 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.
* 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*

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

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

Where:

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

*Client.php: Amendments*

* In Client.php, the items in the menu area are:
* Account Details
* Current Plan
* Current Bookings
* Search
* Discussion

*Provider.php: Amendments*

* In Provider.php, the items in the menu area are:
* Account Details
* Current Bookings
* Services
* Discussion

*Agency.php: Amendments*

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

*Client.php menu time: Account Details*

* In Client.php, when someone clicks “Account Details”:
* Highlight the Account Details menu item
* Display the following information in the Operative Area
* Within the Operative Area, display:
* Row 1:
* Label, named NDISNumberLabel, with text: “NDIS Number”
* Label, named NDISNumberValue, with text: ClientContactTable.ClientNDISNumber
* Row 2:
* Label, named NameLabel, with text: “Name:”
* Textbox, named: FirstNameTextBox, with default text: ClientContactTable.FirstName
* Textbox, named: MiddleNameTextBox, with default text: ClientContactTable.MiddleName
* Textbox, named: LastNameTextBox, with default text: ClientContactTable.LastName
* Row 3:
* Label, named BirthDateLabel, with text: “Date of Birth:”
* Textbox, named: BirthDay, with default text: the day component of ClientContactTable.BirthDate
* Textbox, named: BirthMonth, with default text: the month component of ClientContactTable.BirthDate
* Textbox, named: BirthYear, with default text: the year component of ClientContactTable.BirthDate,
* Row 4:
* Label, named AddressLabel, with text: “Address:”
* Textbox, named: UnitNumberTextBox, with default text: ClientContactTable.UnitNumber
* Textbox, named: StreetNumberTextBox, with default text: ClientContactTable.StreetNumber
* Textbox, named: StreetNameTextBox, with default text: ClientContactTable.StreetName
* Textbox, named: StreetTypeTextBox, with default text: ClientContactTable.StreetType
* Row 5:
* Textbox, named: SuburbTextBox, with default text: ClientContactTable.Suburb,
* Dropbox, named: StateDropBox, with options ACT, NSW, NT, QLD, SA, TAS, VIC, WA, and default chosen option is the value of ClientContactTable.State
* Textbox, named: PostcodeTextBox, with default text: ClientContactTable.Postcode
* Row 6:
* Label, named Email1Label, with text: “Email 1:”
* Textbox, named: Email1TextBox, with default text: ClientContactTable.Email1
* Row 7:
* Label, named Email2Label, with text: “Email 2:”
* Textbox, named: Email2TextBox, with default text: ClientContactTable.Email2
* Row 8:
* Label, named MobilePhoneLabel, with text: “Mobile Phone:”
* Textbox, named: MobilePhoneTextBox, with default text: ClientContactTable.MobilePhone
* Row 9:
* Label, named HomePhoneLabel, with text: “Home Phone:”
* Textbox, named: HomePhoneTextBox, with default text: ClientContactTable.HomePhone
* Row 10:
* Label, named PreferredContactLabel, with text: “Preferred Contact Method:”
* Textbox, named: PreferredContactTextBox, with default text: ClientContactTable.PreferredContact
* Row 11:
* Label, named SecurityQuestionLabel, with text: “Security Question:”
* Textbox, named: SecurityQuestionTextBox, with default text:ClientContactTable.SecurityQuestion
* Row 12:
* Label, named CommencementDateLabel, with text: “Joined ManageCare:”
* Label, named CommencementDateValue, with text: ClientContactTable.CommencementDate
* Row 13:
* Label, named AggregateRatingLabel, with text: “Aggregate Rating:”
* Label, named AggregateRatingLabelValue, with text: ClientContactTable.AggregateRating
* Row 14:
* Label, named Numberof RatingsLabel, with text: “Number of Ratings:”
* Label, named NumberOfRatingsLabelValue, with text: ClientContactTable.NumberOfRatings
* Row 15:
* Button, named UpdateDetailsButton, with caption “Update Details” and “on clicking”:
* Make a text box appear with:
* The caption “Are you sure you want to update your details?”
* Button “yes”
* Button “no”
* 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

* 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:
* Check if any of the information in the boxes has been changed
* If yes, then produce the text box asking for acceptance of updates, with yes and no options
* 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

*Client.php menu time: Current Plan*

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

NDISPlanTable.Client = ClientContactID.value; and

PlanEndDate.value = null

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

*Client.php menu time: Current Bookings*

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

BookingTable.Client = ClientContactID.value; and

DateEnd.value = null

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

*Client.php menu time: Search*

* In Client.php, when someone clicks “Search”:
* Highlight the Search menu item
* Display the following information in the Operative Area
* Within the Operative Area, display the following items
* Row 1:
* Label, named: ServiceLabel, with text: “Service:”
* Dropbox, named: ServiceDropBox, where the options are identified as follows:
* Search each row of the ServicesProvidedTable.
* Include those rows where ServiceProvidedTable.Active = Yes.
* Exclude any duplicate entries in ServicesProvidedTable.Service
* Add the resulting list to the ServiceDropBox, with “Any” as the first entry
* Row 2:
* Label, named: SuburbLabel, with text: “Suburb:”
* Textbox, named: SuburbTextbox, which is blank
* Row 3:
* Label, named SuburbRangeLabel, with text: “Distance:”
* Dropbox, named: SuburbRangeDropbox, with values:
* Any
* 5
* 10
* 15
* Row 4:
* Button, named “SearchButton”, with text: “Search”, and on clicking, calls a search function.
* 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):
* ServicesProvidedTable.Service = ServiceDropBox.value
* ServicesProvidedTable.Suburb = SuburbTextbox.value
* Any suburb that is within the number of kilometres in SuburbRangeDropbox.value from the suburb named in SuburbTextbox.value.
* Display headings:
* Row 1
* Label “Logo”
* Label “Provider”
* Label “Phone”
* Label “Email”
* Label “Website”
* Label “Rating”
* For each returned row, display a row with the following information:
* Row 1+X: for the row where ProviderOrganisationTable.ProviderOrganisationID = ServicesProvidedTable.ProviderID:
* The image stored in the location contained in ProviderOrganisationTable.DisplayLogo
* The value in ProviderOrganisationTable.TradingName
* Note: the wording displaying the name needs to be presented as a hyperlink
* The value in ProviderOrganisationTable.MobilePhone
* Or, if null, ProviderOrganisationTable.WorkPhone
* The value in ProviderOrganisationTable.Email1
* A hyperlink to the website in ProviderOrganisationTable.Website
* Note: clicking on the website hyperlink should open a new page
* The value in ProviderOrganisationTable.AggregateRating
* On clicking the hyperlinked name moves to the Discussion section, with the provider. Store the following information
* ClientContactID
* ProviderContactID relevant to the one selected

*Client: Discussion Area*

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

*Provider: Account Details*

* In Provider.php, when someone clicks “Account Details”:
* Highlight the Account Details menu item
* Display the following information in the Operative Area
* Within the Operative Area, display:
* Row 1:
* Label, named NDISNumberLabel, with text: “NDIS Number”
* Label, named NDISNumberValue, with text: ProviderContactTable.ProviderNDISNumber
* Row 2:
* Label, named NameLabel, with text: “Name:”
* Textbox, named: FirstNameTextBox, with default text: ProviderContactTable.FirstName
* Textbox, named: MiddleNameTextBox, with default text: ProviderContactTable.MiddleName
* Textbox, named: LastNameTextBox, with default text: ProviderContactTable.LastName
* Row 4:
* Label, named AddressLabel, with text: “Address:”
* Textbox, named: UnitNumberTextBox, with default text: ProviderContactTable.UnitNumber
* Textbox, named: StreetNumberTextBox, with default text: ProviderContactTable.StreetNumber
* Textbox, named: StreetNameTextBox, with default text: ProviderContactTable.StreetName
* Textbox, named: StreetTypeTextBox, with default text: ProviderContactTable.StreetType
* Row 5:
* Textbox, named: SuburbTextBox, with default text: ProviderContactTable.Suburb,
* Dropbox, named: StateDropBox, with options ACT, NSW, NT, QLD, SA, TAS, VIC, WA, and default chosen option is the value of ProviderContactTable.State
* Textbox, named: PostcodeTextBox, with default text: ProviderContactTable.Postcode
* Row 6:
* Label, named Email1Label, with text: “Email 1:”
* Textbox, named: Email1TextBox, with default text: ProviderContactTable.Email1
* Row 7:
* Label, named Email2Label, with text: “Email 2:”
* Textbox, named: Email2TextBox, with default text: ProviderContactTable.Email2
* Row 8:
* Label, named MobilePhoneLabel, with text: “Mobile Phone:”
* Textbox, named: MobilePhoneTextBox, with default text: ProviderContactTable.MobilePhone
* Row 9:
* Label, named HomePhoneLabel, with text: “Home Phone:”
* Textbox, named: HomePhoneTextBox, with default text: ProviderContactTable.HomePhone
* Row 10:
* Label, named PreferredContactLabel, with text: “Preferred Contact Method:”
* Textbox, named: PreferredContactTextBox, with default text: ProviderContactTable.PreferredContact
* Row 11:
* Label, named SecurityQuestionLabel, with text: “Security Question:”
* Textbox, named: SecurityQuestionTextBox, with default text:ProviderContactTable.SecurityQuestion
* Row 12:
* Label, named CommencementDateLabel, with text: “Joined ManageCare:”
* Label, named CommencementDateValue, with text: ProviderContactTable.CommencementDate
* Row 13:
* Label, named AggregateRatingLabel, with text: “Aggregate Rating:”
* Label, named AggregateRatingLabelValue, with text: ProviderContactTable.AggregateRating
* Row 14:
* Label, named Numberof RatingsLabel, with text: “Number of Ratings:”
* Label, named NumberOfRatingsLabelValue, with text: ProviderContactTable.NumberOfRatings
* Row 15:
* Button, named UpdateDetailsButton, with caption “Update Details” and “on clicking”:
* Make a text box appear with:
* The caption “Are you sure you want to update your details?”
* Button “yes”
* Button “no”
* 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

* 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:
* Check if any of the information in the boxes has been changed
* If yes, then produce the text box asking for acceptance of updates, with yes and no options
* 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

*Provider: Current Bookings*

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

BookingTable.Provider = ProviderContactID.value; and

DateEnd.value = null

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

*Provider.php: Amendments*

* In Provider.php, the items in the menu area are:
* Account Details
* Current Bookings
* Services
* Discussion

*Provider: Discussion Area*

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

*Provider: Account Details*

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

*Provider: Services*

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

*Provider: List of Current Bookings*

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