Test Plan

System testing

Register Primary Guardian

Test Case Specification

Test case #1

1. Test item(s): Create blank primary guardian with no fields except first name
2. Precondition: First name only contains alphabets (A-Z,a-z), an existing center
3. Input required: First name, Center
4. Test procedure:
    Type in first name of participant
    Click submit
5. Output expected:
    Added to database successfully
    Shows the firstname field as the name that is inputted in the index screen
6. Actual test result:
    Added successfully to database
    Shows the First Name displayed on the index screen

Test case #2

1. Test item(s): Create primary guardian with no children or secondary guardian
2. Precondition: First name and last name only contains alphabets (A-Z,a-z), Phone contains only numeric digits(0-9), no special symbols for every field except email
3. Input required: First name, Last name, email, phone, postal code, language, country, allergy, center
4. Test procedure:
    Type in all of the input specified above
    Click submit
5. Output expected:
    Added to database successfully
    Shows correct data that was inputted by the users in the index screen
6. Actual test result:
    Added successfully to database
    Shows correct data that was inputted by the users in the index screen

Test case #3

1. Test item(s): Create primary guardian with one secondary guardian but no children
2. Precondition: First name and last name only contains alphabets (A-Z,a-z), Phone contains

only numeric digits(0-9), no special symbols for every field except email, first name of secondary guardian
3. Input required: First name, Last name, email, phone, postal code, language, country, allergy, center, first name of secondary guardian
4. Test procedure:
    Type in all of the input specified above
    Add secondary guardian
    Click submit
5. Output expected:
    Added to database successfully
    Shows correct data that was inputted by the users in the index screen
    Secondary guardian count = 1
    Children count = 0
6. Actual test result:
    Added successfully to database
    Shows correct data that was inputted by the users in the index screen
    Secondary guardian count = 1
    Children count = 0

Test case #4

1. Test item(s): Create primary guardian with multiple secondary guardian but no children
2. Precondition: First name and last name only contains alphabets (A-Z,a-z), Phone contains only numeric digits(0-9), no special symbols for every field except email, multiple names of secondary guardians
3. Input required: First name, Last name, email, phone, postal code, language, country, allergy, center, multiple secondary guardian names
4. Test procedure:
    Type in all of the input specified above
    Add two secondary guardians
    Click submit
5. Output expected:
    Added to database successfully
    Shows correct data that was inputted by the users in the index screen
    Secondary guardian count = 2
    Children count = 0
6. Actual test result:
    Added successfully to database
    Shows correct data that was inputted by the users in the index screen
    Secondary guardian count = 2
    Children count = 0

Test case #5

1. Test item(s): Create primary guardian with one children but no secondary guardian
2. Precondition: First name and last name only contains alphabets (A-Z,a-z), Phone contains only numeric digits(0-9), no special symbols for every field except email, first name of child
3. Input required: First name, Last name, email, phone, postal code, language, country, allergy, center, first name of child
4. Test procedure:
    Type in all of the input specified above
    Add one child
    Click submit
5. Output expected:
    Added to database successfully
    Shows correct data that was inputted by the users in the index screen
    Secondary guardian count = 0
    Children count = 1

6. Actual test result:
    Added successfully to database
    Shows correct data that was inputted by the users in the index screen

Test case #6

1. Test item(s): Create primary guardian with multiple children but no secondary guardian
2. Precondition: First name and last name only contains alphabets (A-Z,a-z), Phone contains only numeric digits(0-9), no special symbols for every field except email, first name of multiple children
3. Input required: First name, Last name, email, phone, postal code, language, country, allergy, center
4. Test procedure:
    Type in all of the input specified above
    Add two children
    Click submit
5. Output expected:
    Added to database successfully
    Shows correct data that was inputted by the users in the index screen
    Secondary guardian count = 0
    Children count = 2

6. Actual test result:
    Added successfully to database
    Shows correct data that was inputted by the users in the index screen
    Secondary guardian count = 0
    Children count = 2

Test case #7

1. Test item(s): Create primary guardian with one children and one secondary guardian
2. Precondition: First name and last name only contains alphabets (A-Z,a-z), Phone contains only numeric digits(0-9), no special symbols for every field except email, first name of secondary guardian and child
3. Input required: First name, Last name, email, phone, postal code, language, country, allergy, center
4. Test procedure:
    Type in all of the input specified above
    Add one secondary guardian
    Add one child
    Click submit
5. Output expected:
    Added to database successfully
    Shows correct data that was inputted by the users in the index screen
    Secondary guardian count = 1
    Children count = 1

6. Actual test result:
    Added successfully to database
    Shows correct data that was inputted by the users in the index screen
    Secondary guardian count = 1
    Children count = 1

Test case #8

1. Test item(s): Create primary guardian with multiple children and secondary guardian
2. Precondition: First name and last name only contains alphabets (A-Z,a-z), Phone contains only numeric digits(0-9), no special symbols for every field except email
3. Input required: First name, Last name, email, phone, postal code, language, country, allergy, center, multiple first name of children and secondary guardian
4. Test procedure:
    Type in all of the input specified above
    add 2 secondary guardian
    add 2 children
    Click submit
5. Output expected:
    Added to database successfully
    Shows correct data that was inputted by the users in the index screen
    Secondary guardian count = 2
    Children count = 2
6. Actual test result:
    Added successfully to database

Shows correct data that was inputted by the users in the index screen
Secondary guardian count = 2
Children count = 2

1. **Test case #9**

1. Test item(s): check for invalid input field
2. Precondition: input fields are blank, or it is in valid form in the edit page
3. Input required: First Name, Last Name, Phone Number, Email, Birthdate, Country, Language
4. Test procedure:

   Type in special character or symbols into all of the input specified above
   Click Submit

5. Output expected:

   invalid input fields are highlighted with red and error message will be shown beside it.
   system does not submit the form

6. Actual test result:

   invalid fields were correctly showed up with highlighted color, and error message showed up
   form did not submitted

Test case #10

1. Test item(s): delete Primary Guardian
2. Precondition: Primary Guardian and necessary information is displayed
3. Input required: None
4. Test procedure:

   click delete button once primary guardian detail shows up

5. Output expected:

   primary guardian is correctly deleted and deleted from the index page

6. Actual test result:

   primary guardian was correctly deleted from database

Drop-in Session Sign In Sheet

Test case #1 : valid primary guardian and Drop-in Session (VALID CASE)

1. Test item(s) : Create a sign in.
2. Create a Primary-Guardian
3. Go to “Sign” in and search for the primary guardian that you just created.
4. Chose a Drop-in Session for a given date.

Expected : Successfully added popup box.

5. Go to : admin home ⇒ generate report.
6. Click “Submit”
7. **Expected :** You should be able to see the newly associated primary guardian and drop session in the table.

Test case #2 : invalid primary guardian (INVALID CASE)

1. Test items(s) : Create sign in
2. Preconditions : No primary guardian created in system / valid drop-in sessions.
    a. Go to “Sign in”
    b. Search for any primary guardian.
    c. **Expected :** no primary guardian record should show up because no records exists.

Test Case #3 : Delete Session

1. Test item(s) : deleting a session.
2. Precondition : the session to be deleted exists.
    a. Go to drop-in session index page.
    b. Click on “delete” for a session that you want to delete.
    c. **Expected :** the deleted session should not be visible or searchable in the system.
    Expected 2: any families who attended those sessions should have their record deleted as well.

Check-Out Library Items

Test Case #1 : valid library items and valid user.

1. Precondition : a library item exists and the user has not checked out library items over his/her limits (2).
    a. Navigate to “Library Check-Out”
    b. Select the valid user and a valid library items from the dropdown.
    c. **Expected Result :** both items should be selectable from the dropdown.
    d. Select a due date in the future.
    e. Click create, the library checkout items should not be displayed in the table.

Test case #2 : Invalid inputs

1. Precondition : insert invalid inputs during the library item check out
    a. Navigate to “Library Check-Out”
    b. Try to check out an item for a primary guardian but using a return date that is past the current date.
    c. **Expected Result :** red text should pop up to indicate invalid date input; cannot have a return date that is back in time.
2. Precondition : no input for return date
    a. Navigate to “Library Check-Out”
    b. Try to check out an item for a primary guardian but do not input any return date field.
    c. **Expected Result :** red text should pop up to indicate invalid date input; cannot have a empty return date.

Test case #3 : Checking in a library item.

1. Precondition : the library item to be checked in must be currently checked out
    a. Go to “Library Checkout”
    b. Select a checkout items
. c. Change status to check in
    d. **Expected Result :** the library item is not checked in and can be further checked out again.

Test case #4 : Deleting Library Items.

1. Go to library item index
2. Delete a particular library items.
3. **Expected Result** : the deleted item should not be searchable or check out anymore.

Create Centers

1. Test case #1 (Valid)

2. Test item(s): Create an Center
3. Precondition: Valid Phone number is entered, only numbers are allowed, valid name is entered
4. Input required: A valid phone number, valid name
5. Test procedure:
    Type in a valid number without character, and valid Name without number or symbols
    Click submit
6. Output expected:
    Added to database successfully
    Shows correct data that was inputted by the users in the index screen
7. Actual test result:
    Added successfully to database
    Shows correct data that was inputted by the users in the index screen

2. Test Case #2 (INVALID INPUT)

1. Test items(s) : Creating a Center but invalid inputs
2. Precondition : any invalid or a combination of invalid input fields.
    a. Go to “Create Center”
    b. Input invalid phone field(with alphabets)
    c. **Expected :** Alert letter in red indicating fields being filled in invalidly.

Edit Centers

Test case #2

1. Test item(s): Edit an Center
2. Precondition: Valid Phone number is entered, only numbers are allowed, valid name is entered
3. Input required: A valid phone number, valid name
4. Test procedure:
    Type in a valid number without character, and valid Name without number or symbols
    Click submit
5. Output expected:
    Edited to database successfully
    Shows correct data that was inputted by the users in the index screen
6. Actual test result:
    Edited successfully to database
    Shows correct data that was inputted by the users in the index screen

Generate Report

Test case #1

1. Test item(s): Generate report (view in HTML)
2. Precondition: Start date and End date are specified using the format: dd/mm/yyyy
    Start date cannot be later than end date
3. Input required: Start date and End date
4. Test procedure:
    Type in (or pick from date picker)Start date and End date
    Click generate report
5. Output expected:
    Three tables:
    1. Contained information of number of new parents, children, visit secession for each center.
    2. Contained information of how many different languages are the parents speaking in each center.
    3. Contained the culture background information of parents in each center

6. Actual test result:
Three tables with correct information displayed in the generate view.

Test case #2

1. Test item(s): Generate report (export CSV)
2. Precondition: Start date and End date are specified using the format: dd/mm/yyyy
Start date cannot be later than end date
3. Input required: Start date and End date
4. Test procedure:
Type in (or pick from date picker)Start date and End date
pick the CSV radio button
Click generate report
5. Output expected:
A CSV file with three tables is downloaded:
1. Contained information of number of new parents, children, visit secession for each center.
2. Contained information of how many different languages are the parents speaking in each center.
3. Contained the culture background information of parents in each center
6. Actual test result:
Three tables with correct information is downloaded in a CSV file.

Test case #3

1. Test item(s): Generate report (export Excel)
2. Precondition: Start date and End date are specified using the format: dd/mm/yyyy
Start date cannot be later than end date
3. Input required: Start date and End date
4. Test procedure:
Type in (or pick from date picker)Start date and End date
pick the Excel radio button
Click generate report
5. Output expected:
An Excel file with three tables is downloaded:
1. Contained information of number of new parents, children, visit secession for each center.
2. Contained information of how many different languages are the parents speaking in each center.
3. Contained the culture background information of parents in each center
6. Actual test result:
Three tables with correct information is downloaded in an Excel file.

Test case #4

1. Test item(s): Generate report with start date later than end date
2. Precondition: Start date and End date are specified using the format: dd/mm/yyyy
   Start date cannot be later than end date
3. Input required: Start date later than End date
4. Test procedure:
   Type in (or pick from date picker)Start date and End date
   pick the View in HTML button
   Click generate report
5. Output expected:
   Three tables are shown in HTML with empty cells:
6. Actual test result:
   Three tables with correct information are shown in HTML.

Test case #5

1. Test item(s): Check visit history
2. Precondition: Start date and End date are specified using the format: dd/mm/yyyy
   Start date cannot be later than end date
   An primary guardian is selected
3. Input required: Start date and End date
   Choose an primary guardian
4. Test procedure:
   Type in (or pick from date picker)Start date and End date
   Search for and pick a primary guardian.
   Click Check History
5. Output expected:
   An HTML page with check in history of the chosen primary guardian is shown.
6. Actual test result:
   The correct check in history is shown.

Resources Test

Test Case #1: Positive stock smaller than Current Stock

1. Log in as Staff if not already.
2. Click Resources on the navigation panel.
3. Create a New Resource:
   a. Number Available: 10
   b. Name: Test
4. After clicking “Create”, verify that a new resource “Test” exists in the list with Current Stock = 10.

5. Enter “9” for resource “Test” and click save.
6. You should see the “Edit Successful” page. Click “Go to the Resource Page”.
7. You should see that resource “Test” has Current Stock = 9 and Total Handed Out = 1.

Test Case #2: Positive stock bigger than Current Stock

1. Continued from Test Case #1, Enter “10” for resource “Test” and click save.
2. You should see the “Edit Successful” page. Click “Go to the Resource Page”.
3. You should see that resource “Test” has Current Stock = 10 and Total Handed Out = 1.

Test Case #3: Negative stock(number handed out)

1. Continued from Test Case #2, Enter “-5” for resource “Test” and click save.
2. You should see the “Edit Successful” page. Click “Go to the Resource Page”.
3. You should see that resource “Test” has Current Stock = 5 and Total Handed Out = 6.

Test Case #4: Invalid Stock

1. Make the stock field blank and save.
2. Go back to the edit page and note that no changes are made.
3. Enter some non-number text in stock field and save.
4. Go back to the edit page and note that no changes are made.

Test Case #5: Delete a Resource

1. Continued from Test Case #4, click Delete beside “Test”
2. Confirm the delete
3. You should see that “Test” no longer exists.

Referrals Test

Test Case #1: Create referrals

1. Log in as Staff if not already.
2. Click Referrals on the navigation panel.
3. Create a New Referral:
    a. Name: Test
4. After clicking “Create”, verify that a new referral “Test” exists in the list.

Test Case #2: Edit referrals

1. Continued from Test Case #1.
2. Enter “9” for referral“Test” and click save.
3. You should see the “Edit Successful” page. Click “Go to the Referrals Page”.
4. You should see that referral “Test” has Total Referrals Made = 9.

Test Case #3: Invalid Count

5. Make the stock field blank and save.
6. Go back to the edit page and note that no changes are made.

7. Enter some non-number text in stock field and save.
8. Go back to the edit page and note that no changes are made.

Test Case #4 : Delete referrals

1. Go to Referrals index.
2. Select the referral you want to delete.
3. Click “delete”
4. **Expected :** the deleted referral should not be visible anymore.

Library Items Test

Test Case #1: Create Library Items

1. Log in as Staff if not already.
2. Click Library the navigation panel.
3. Create a New Library Item:
    a. Broken: No
    b. Value of Item: 30
    c. Note: Test Item
    d. Name: Test
    e. ItemType: Video
    f. Sanitized: Yes
4. After clicking “Create”, verify that a new Library item “Test” exists in the list and values for the various columns match those entered in step 3. Ignore values for other columns.

Test Case #2: Edit Library Items

1. Continued from Test Case #1.
2. Click “Edit” beside the “Test” library item
3. Change Sanitized to “No” and click “Save” button.
4. After clicking “Save”, verify that Library item “Test” exists in the list and values for the “Sanitized” column is “No”

Test Case #3: Delete Library Items

1. Continued from Test Case #2.
2. Click “Delete” beside the “Test” library item
3. Confirm the delete.
4. After the confirmation, verify that Library item “Test” is absent in the list.

Test Case #4: Invalid Case

1. Continued from Test Case #3.
2. Create a New Library Item with every field empty.
3. All the field for library items are nullable. So after clicking “Create”, the Library Item should be created.

Special Event Test

Test Case #1: Create Special Event

1. Log in as Staff if not already.
2. Click Special Event on the navigation panel.
3. Create a New Special Event:
    a. Name: Testing
    b. Date: 04/04/2013
4. After clicking “Create”, verify that a new Special Event “Test” exists in the list with the right date.

Test Case #2: Edit Special Event

1. Continued from Test Case #1.
2. Click on edit link of test
3. change the name to “test1”, date to 03/04/2014 and click Save.
4. You should see the “Edit Successful” page. Click “Go to the Special Event Page”.
5. You should see that Special Event “Test1” has the date 03/04/2014.

Test Case #3: Delete Special Event

6. Continued from Test Case #1.
7. Click on delete link of test
8. You should see that the special event is deleted.

Test Case #4: Invalid case

1. Log in as Staff if not already.
2. Click Special Event on the navigation panel.
3. Create a New Special Event with empty name and date.
4. After clicking “Create”, the error message should show up.

Integration testing

Center Admin Relationship

1. Test case ID #1
2. Name of components being integrated and tested : Center and Admin user
3. Test duration: ~3hours
4. Test date: March 26 2013
5. Name of tester(s) : Kevin, Steven, Peter, Ante
6. Actual results of test (you may use the test plan as result recording mechanism)
    - Center was correctly created and Admin user was able to go to that center through the Admin Navigation bar

PrimaryGuradian and Child relationship

1. Test case ID #2
2. Name of components being integrated and tested : PrimaryGuardian and Child
3. Test duration: ~30hours
4. Test date: March 23 2013
5. Name of tester : Kevin, Peter
6. Actual results of test (you may use the test plan as result recording mechanism)
    - Children were correctly added and associated to the Primary guardian

PrimaryGuradian and Secondary Guardian

1. Test case ID #3
2. Name of components being integrated and tested : PrimaryGuardian and Seconday Guardian
3. Test duration: ~30hours
4. Test date: March 24 2013
5. Name of tester : Kevin, Peter
6. Actual results of test (you may use the test plan as result recording mechanism)
    - Secondary Guardian was correctly added if specified during the creation of a primary guardian. This can be verified during a name search for a secondary guardian.

Admin and Generate Report Relationship

1. Test case ID #4
2. Name of components being integrated and tested : Admin and Generate Report
3. Test duration: ~20hours
4. Test date: March 23 2013
5. Name of tester : Kevin, Steven, Rufus, Peter
6. Actual results of test (you may use the test plan as result recording mechanism)
    - Report were created with the time range specified by a Admin User in HTML, CSV, and Excel File

Admin and Generate Report Relationship

1. Test case ID #5
2. Name of components being integrated and tested : Admin and Check History
3. Test duration: ~20hours
4. Test date: March 23 2013
5. Name of tester : Kevin, Steven, Rufus, Peter
6. Actual results of test (you may use the test plan as result recording mechanism)
    - Number of visits displayed correctly to the given primary guardians

Admin and Add Staff Account

1. Test case ID #6
2. Name of components being integrated and tested : Admin and Add Staff Account
3. Test duration: ~10hours
4. Test date: March 22 2013
5. Name of tester : Kevin, Steven, Peter, Ante
6. Actual results of test (you may use the test plan as result recording mechanism)
    - Staff account was created and account was saved successfully into database, newly created staff account was able to log in

Center and Staff Relationship

1. Test case ID #7
2. Name of components being integrated and tested : Center and Staff
3. Test duration: ~20hours
4. Test date: March 23 2013
5. Name of tester : Steven, Ante, Peter
6. Actual results of test (you may use the test plan as result recording mechanism)
    - Admin creating a new staff and assigning they account to a center; upon logging into that staff account, that staff can only perform functions, and data access pertaining to that center.

Sign in and Center Relationship

1. Test case ID #8
2. Name of components being integrated and tested : Sign in and center association
3. Test duration: ~20hours
4. Test date: March 23 2013
5. Name of tester : Steven
6. Actual results of test (you may use the test plan as result recording mechanism)
    - When an admin or staff signs in a family to a session at a particular center, that center will now have recorded the family's attendance. This can be verified in the generated reported or in the user tracking functionalities.

Library Item and Center Relationship

1. Test case ID #9
2. Name of components being integrated and tested : Library and Center
3. Test duration: ~10hours
4. Test date: March 21 2013
5. Name of tester : Kevin, Steven
6. Actual results of test (you may use the test plan as result recording mechanism)
    - Library items created in a particular center can be only be associated with that center; in other words, a toy created in center A will not be accessible by staff in center B.

Library Check out and Centre Relationship

1. Test case ID #10
2. Name of components being integrated and tested : Center and Library Check out
3. Test duration: ~20hours
4. Test date: March 14 2013
5. Name of tester : Kevin, Steven
6. Actual results of test (you may use the test plan as result recording mechanism)
   - Admin navigate to or staff log on to a certain centre and go to the Library check out page. The user should only able to see the library items associate to that centre. And the user should also only able to check out the library item associated with that centre.

Special Event and Centre Relationship

1. Test case ID #11
2. Name of components being integrated and tested : Center and Special Event
3. Test duration: ~10hours
4. Test date: March 25 2013
5. Name of tester : Steven
6. Actual results of test (you may use the test plan as result recording mechanism)
   - Admin navigate to or staff log on to a certain centre and go to the special event page. The user should only able to see the special events associate to that centre. And when user creates a special event, that special event should associated with that centre.

Resources and Center Relationship

1. Test case ID #12
2. Name of components being integrated and tested :Center and Resources
3. Test duration: ~10hours
4. Test date: March 21 2013
5. Name of tester : Kevin, Steven, Ante, Peter
6. Actual results of test (you may use the test plan as result recording mechanism)
   - Resources were created and were saved to that center

Primary Guardian and Center Relationship

1. Test case ID #13
2. Name of components being integrated and tested : Primary Guardian & Center.
3. Test duration: ~10hours
4. Test date: March 21 2013
5. Name of tester : Kevin, Steven, Peter
6. Actual results of test (you may use the test plan as result recording mechanism)
   - A primary guardian that was created will be associated to the center at which he/she physically attended during his/her initial registration.

Referrals and Center Relationship

1. Test case ID #14
2. Name of components being integrated and tested :Center and Referrals
3. Test duration: ~10hours
4. Test date: March 21 2013
5. Name of tester : Ante
6. Actual results of test (you may use the test plan as result recording mechanism)
    - Referrals created were correctly associated to the center

Library Item and Library Borrower Relationship

1. Test case ID #15
2. Name of components being integrated and tested : Library and Primary Guardian Borrower
3. Test duration: ~10hours
4. Test date: March 21 2013
5. Name of tester : Kevin, Steven
6. Actual results of test (you may use the test plan as result recording mechanism)
    - Library items that were checked out should not be able to be checked out again until the item is returned by the person who borrowed it
    - The checked out status in the library items tab must be coherent with the checkout status in library check out tab

Unit testing

The unit test project documents the unit test plan. The project uses the unit test framework built into Visual Studio 2010 referred to simply as the Microsoft Unit Test Framework. The main pieces that need to be unit tested are the of “Models” and the “Controllers” of the Model View Controller (MVC) architecture. The unit test files can be found in their respective folders in the Test project.

The unit test names will have a verbose naming scheme such that a reader of the unit test can understand the purpose of the test. For example inside a file called HomeControllerTest.cs in the Controllers folder of the test project, there may be a test called Index_NotLoggedIn_ShouldRedirectToLogin. A reader of this test can expect the test to verify that the controller will redirect a user that is not logged in to the login page.

As the application makes extensive use of the database, the database access must be decoupled from the controllers to allow test runs to complete quickly. To accomplish this, the main project makes use of a runtime dependency injection framework called Ninject that injects Interface dependencies into the controllers.

The interfaces and concrete classes that implement these interfaces can be found in the Repositories folder in the project. For example, there is the interface ICenterRepository.cs and the concrete class CenterRepository.cs. There is one more interface that wraps all the repositories into one class called IRepositoryService.cs and the concrete class is called DbRepositoryService.cs. Every controller accesses the database (repository) through

IRepositoryService.cs. A similar method is used to inject user authentication dependencies that call static methods to authenticate users. These layers that depend on an actual data repository such as a database are not unit tested but are verified through system and integration testing.

Since data access depends on an interface and not a database, the data repositories' interfaces can be mocked and passed into the controllers through their constructors. The mocking framework chosen is Moq. The controllers are instantiated with these mocks and results of Controller Actions (methods) are verified with the unit test framework.

Model validation is much simpler using the model data annotations and a method defined in the Controller classes, ValidateModel. This sets the ModelState of a controller and is verified to be valid.

Asserts are used throughout the unit tests. After a test run, the Microsoft Unit Test Framework produces a test result summary and failed tests are marked. If an assert failed, the framework notes the expected value and the actual value. The test also notes If the test failed for another reason, for example, a null pointer exception. You can find the exact line of code that produced the error.