

Translational Research Web Application Framework

by

Nikolay S. Nepochatov

A CAPSTONE PROJECT

Presented to the Department of Medical Informatics &

Clinical Epidemiology

and the Oregon Health & Science University

School of Medicine

in partial fulfillment of

the requirements for the degree of

Master of Biomedical Informatics

May 2009

School of Medicine
Oregon Health & Science University

CERTIFICATE OF APPROVAL

This is to certify that the Master's Capstone Project of

Nikolay S. Nepochatov

“Translational Research Web Application Framework”

Has been approved

Dr. Bill Hersh

Table of Contents

Abstract.....	4
Introduction.....	5
Development.....	7
Study example.....	7
Database.....	9
Application.....	17
Application structure.....	17
Model-View-Controller structure of the application.....	19
Conclusion.....	24
Appendix A. Database creation script.....	25
Appendix B. Database population script.....	30
Bibliography.....	55

Abstract

An important service that the Oregon Clinical and Translational Research Institute (OCTRI) provides to its investigators is database application development. For the typical investigator-initiated clinical research study, OCTRI uses standardized database structures and application code developed in over a decade of supporting such projects. The database structures are based on the Clinical Data Interchange Standards Consortium Specification for the Operational Data Model (CDISC ODM) data model and include structures that hold study metadata. The study metadata defines the study structure in terms of study events (e.g., inpatient admissions, outpatient visits, phone interviews, etc), patients, and case report forms. Study metadata are used to generate a complete study calendar and e-case report form records for a patient based on study entry date. The system consists of a FirebirdSQL database and an application implemented in Borland Delphi. Firebird is an open source relational database management system that was released in March, 2002. Borland Delphi is a strong typed object oriented programming language that was introduced in 1996. The purpose of the Translational Research Web Application Framework project is to port the existing system into the more popular MySQL relational database management system paired with the un-typed scripting language called PHP. The PHP code takes advantage of the open source ZEND Framework, which facilitates many of the common functions and interactions with the MySQL database, making the code easy to read and understand, and the coding process very efficient and fast. The scope of the project includes an application structure and a Model View Controller (MVC) framework that will be used by other developers who will create a graphical user interface for their applications.

Translational Research Web Application Framework

Introduction

Oregon Clinical and Transitional Research Institute (OCTRI) is a unique partnership between Oregon Health & Science University and Kaiser Permanente's Center for Health Research. (Orwoll) Primary aims of OCTRI include the development of an academic "home" that is specifically devoted to clinical and translational research, the implementation of coordinated infrastructure of core research tools, and training of interdisciplinary investigators through a strong, diverse educational curriculum. (Orwoll)

One of the research tools OCTRI provides investigators with is database application development. For the typical investigator-initiated clinical research study, OCTRI uses standardized database structures and application code developed in over a decade of supporting such projects. The database structures are based on the CDISC ODM data model and include structures that hold study metadata. The study metadata defines the study structure in terms of study events (e.g., inpatient admissions, outpatient visits, phone interviews, etc), patients, and case report forms. Study metadata are used to generate a complete study calendar and e-case report form records for a patient based on study entry date.

A typical study this system supports might be a Parkinson's researcher who wants to explore the effects of an adjunct drug on the efficacy of L-Dopa in controlling Parkinson's symptoms. The study calls for a 4-day inpatient admission with a variety of measures of Parkinsonism taken every 0.5 hour from 7:00AM to 6:00PM each day under various dosing levels and timings of the adjunct drug. There are 4 different assessments that are performed each 0.5 hr. and each assessment includes multiple data items. Upon completing the metadata, a researcher will be able to enroll a participant into a study and generate a complete schedule of interventions as well as record results of the study. Because the system makes use of the relational database, researchers will be able to build reports of collected data.

The current system is built with the support of an open source relational database management system called Firebird and a strong-typed object oriented programming language called Delphi. Both of these products originated from a company called Borland in the 1990s. (Zarko Gajic) (InterBase/Firebird Documentary Team) This project's main purpose is to port the existing system to the more popular MySQL database management system and un-typed scripting language called PHP. MySQL and PHP presence is much more pronounced on the World Wide Web than Firebird and Delphi, which means that many more resources are available for MySQL and PHP developers. One such resource is Zend Framework. The Zend Framework is a collection of PHP classes that provide a lightweight, loosely-coupled component library simplified to provide 80% of the functionality needed by most of the web 2.0 applications. (Zend Technologies Ltd) Framework takes advantage of Model View Controller (MVC) architectural pattern that isolates business logic from the user interface. Such a pattern

provides for ease of modification of either the Graphical User Interface or the business logic without affecting one another. (Microsoft Corporation)

In this project, the creation of any kind of user Interface resides outside of the scope of the project; however, the existence of the Model View Controller pattern is very important for the future development of the application.

Development

Study example

Consider the following study provided by Robert Schuff, Director of Clinical Research Informatics in OCTRI.

Case study Summary. "Neurocognitive effects of Hypothyroidism"

Scientific abstract. *The effects of disordered thyroid function on the developing brain are well known. However, less is understood about the effects of thyroid dysfunction on cognition in the mature brain, particularly regarding effects of mild thyroid disease, which is common in the aging population. The goal of this study is to identify which neural systems and cognitive domains are affected by subclinical hypothyroidism (SCH) in adulthood. Our hypothesis is that SCH causes specific defects in long-term memory, which are worse with aging. The study will recruit healthy subjects receiving l-thyroxine for hypothyroidism who have normal TSH levels. In order to assess the effects of aging, subjects with broad range of ages (20 to 80 years) will be studied. In a placebo-controlled, blinded, crossover fashion, subjects will receive either their usual dose of l-thyroxine or a lower dose calculated to induce SCH for 3 months. Subjects will undergo*

specific neurocognitive tests chosen to detect subtle deficits in short- and long- term memory, and compare those to tests that are not expected to be affected.

Study arm. Study the effect of subclinical hypothyroidism on specific measures of long-term memory.

Study Design. Two groups (young with hypothyroidism, old with hypothyroidism) with two treatment arms: euthyroid and SCH. The order of treatment (euthyroid vs. SCH) will be randomized. The following table represents a detailed schedule of the study.

Detailed schedule of events								
event	action	Screening	Baseline	Week 6	Week 12	Week 18	Week 24	Week 30
CLINICAL	Height	X						
	Weight	X	X	X	X	X	X	X
	BP	X	X	X	X	X	X	X
CBC	WBC	X						
	Hc	X						
	Hg	X						
	Platelet	X						
COGNITIVE	SOP6		X		X		X	
	SOP8		X		X		X	
	SOP10		X		X		X	
	SOP12		X		X		X	
MOOD	POMSA		X		X		X	
	POMSC		X		X		X	
	POMST		X		X		X	
	POMSD		X		X		X	
	POMSF		X		X		X	
	POMSV		X		X		X	
LIPIDS	LDL	X		X		X		
	Trig	X		X		X		
THYROID	TSH	X	X	X	X	X	X	X
	T3		X	X	X	X	X	
	FT4		X	X	X	X	X	

Database

The database for this project consists of 19 tables connected with one-to-many relationship links. It can logically be divided into 2 parts: the metadata and the participant data. The metadata part, as the name implies, contains information about the structure of the study, whereas the participant part contains a detailed schedule of each participant/patient enrolled in the study. Now that we have an example of a study, let us look at the database a little closer.

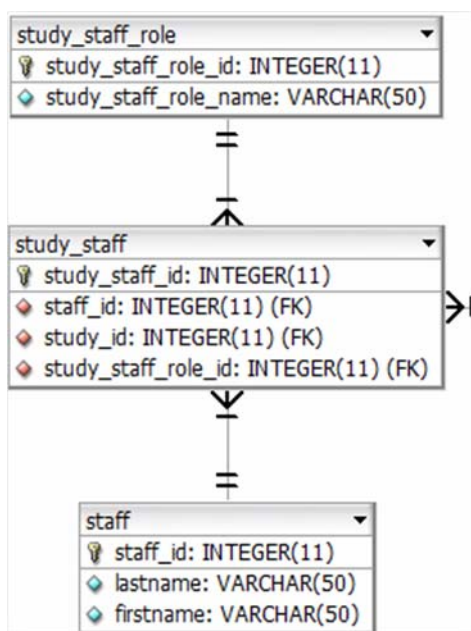


Figure 1: Metadata. Staff Information

An intricate part of every study is of course staff information. These tables contain information about the entire staff of the study. The **STAFF** table contains first name and last name of every person that is performing some function in a study. The **STUDY_STAFF_ROLE** table contains all of the roles a study has (for example: Principal investigator, Sub-investigator, Coordinator and so on.). The **STUDY_STAFF** table combines two previously mentioned tables and connects each person with a particular study role he/she will be

playing. This structure allows for each person to occupy more than one role in the study and it also allows multiple persons to be assigned the same role, in other words there can be several coordinators per study, and the same person can be a sub-investigator as well as a coordinator.

The next portion of the metadata is the study itself. The **STUDY** table contains the study name and its shortened title. In our example, the study name would be “The effect of subclinical hypothyroidism on specific measures of long-term memory” and the study title would be “Neurocognitive effects of Hypothyroidism”. The **STUDY** table is connected with the **ARM** table with a one-to-many relationship, which means that each study can have one or more arms. In our case we have two arms which are

“euthyroid” and “SCH”. The **ARM** table contains an is_active column which allows for easy activation or deactivation of the study arm, it also contains the name of an arm

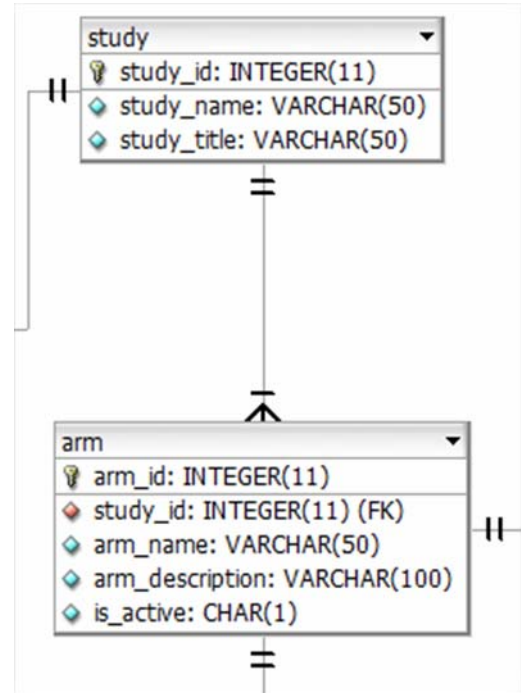


Figure 2: Metadata. Study and study arm

and a short description of it.

Figure 3 shows the relationship of the **STAFF** with the **STUDY**. After we assigned a person to a specific study role, we assign a study to the established relationship.

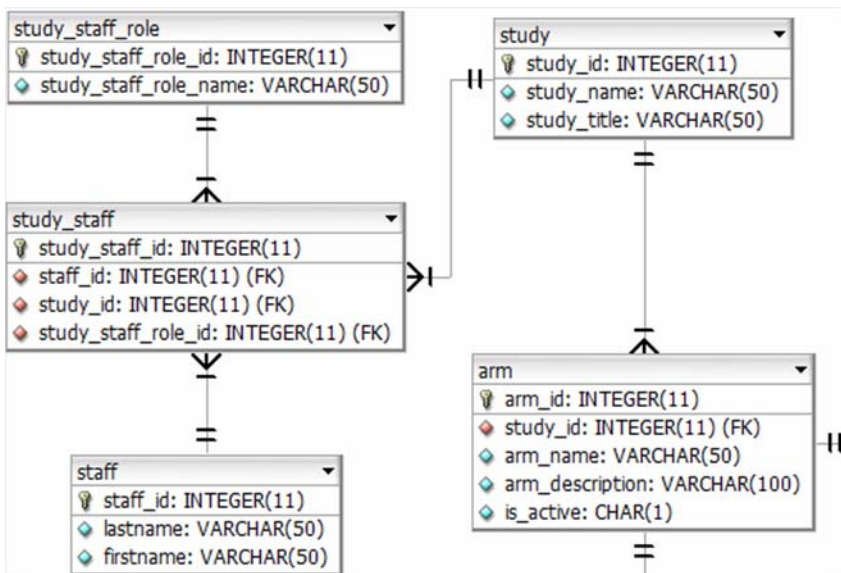


Figure3: Metadata. Staff and study

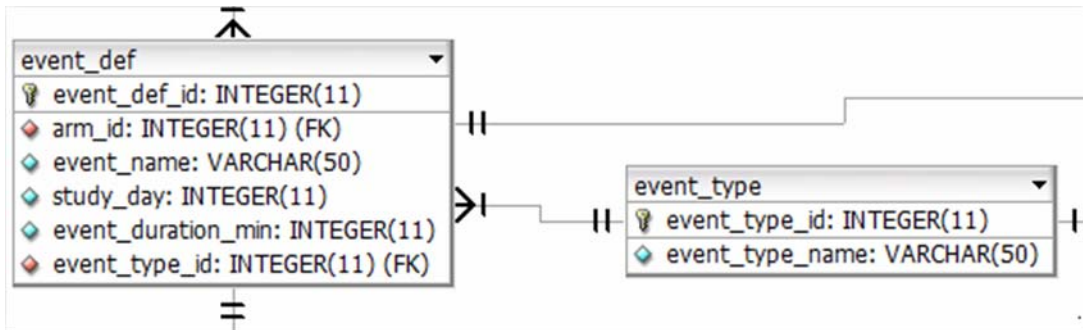


Figure 4: Metadata. Event definition and Event type

The next part of the metadata is the **EVENT_DEF** and **EVENT_TYPE** tables. The **EVENT_DEF** table is an event definition table and contains key information about each event of the arm of the study. In our example we have the following seven events: screening, baseline, week 6, week 12, week 18, week 24, and week 30. Each of these events has a name, day of the study (Screening event having day 0, Baseline – day 20, Week 6 – day 62, and so on), duration in minutes, and a type of event. All event types reside in the **EVENT_TYPE** table, which is connected to the **EVENT_DEF** table with a many-to-one relationship meaning that each of the events can have only one type (for example “Outpatient visit”, “phone interview”, or “inpatient visit”). In our case all events

will be one type – an outpatient visit.

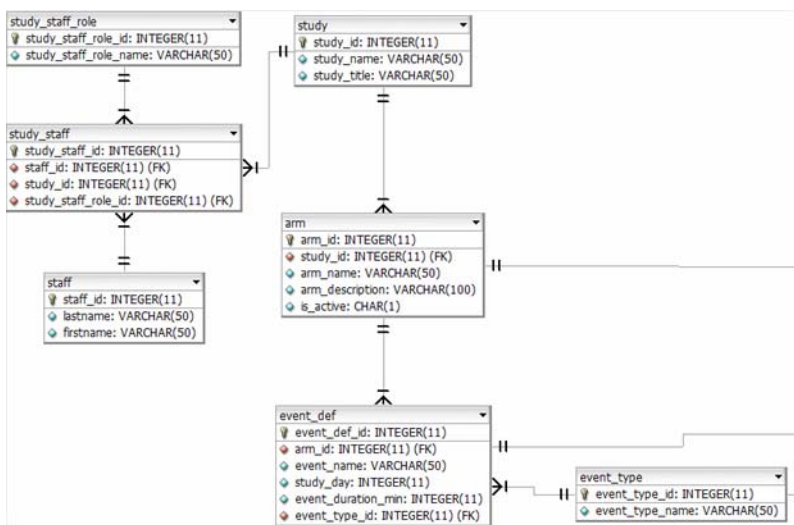


Figure 5: Metadata so far

Figure 5 shows the metadata relationships so far. We can see how **ARM** is connected to an **EVENT_DEF** table with a one-to-many relationship, meaning that each study arm can have

many events (like in our example we have a euthyroid and SCH arms each having seven previously described events).

Figure 6 shows the next hierarchy of our database structure, in a particular **TIMEPOINT_DEF** table and its relationship with **EVENT_DEF** table. In this structure each event (Baseline, Week 6, and so on) can have one or more

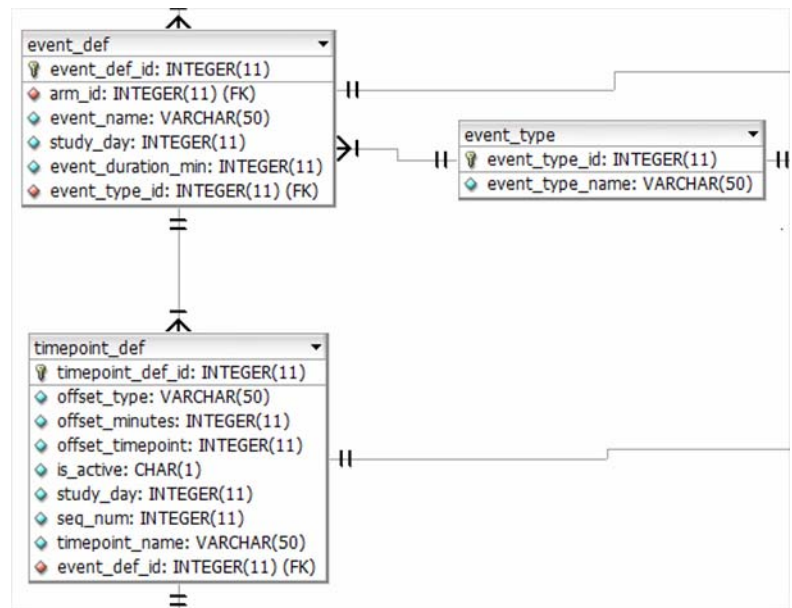
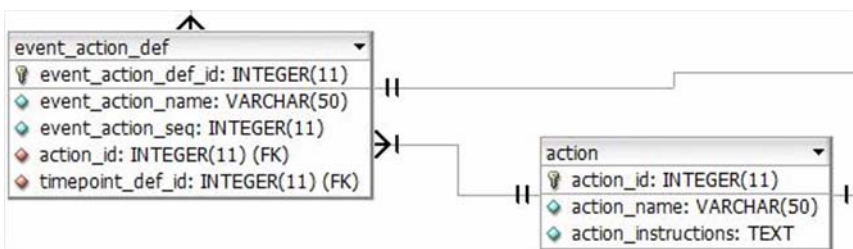


Figure 6: Metadata. Timepoint definitions

timepoints associated with it. In our example, each timepoint is represented by the type of measurement we are interested in collecting. Looking at the detailed schedule of events for our example study we can identify 6 timepoints for each event. They are "Clinical" measurements "CBC", "Cognitive", "Mood", "Lipids", and "Thyroid". Each timepoint has a name, offset values that are used in calculations of the actual participant schedule, active flag, day of the study the timepoint has to be performed, and a sequence number of the timepoint relative to other timepoints.



The last piece of the metadata represented in the Figure 7 is the action itself.

Figure 7: Metadata. Event actions

Each timepoint can have one or more actions associated with it. An action is the actual test that researched will perform on a participant. Our example study contains quite a few actions (Height, Weight, and BP are the examples of the actions). Actions themselves reside in a separate **ACTION** table that contains action name and the instruction for an investigator on how to perform the action. Each action is assigned a sequence number which tells an investigator the order the action must be performed relative to other actions of the same timepoint.

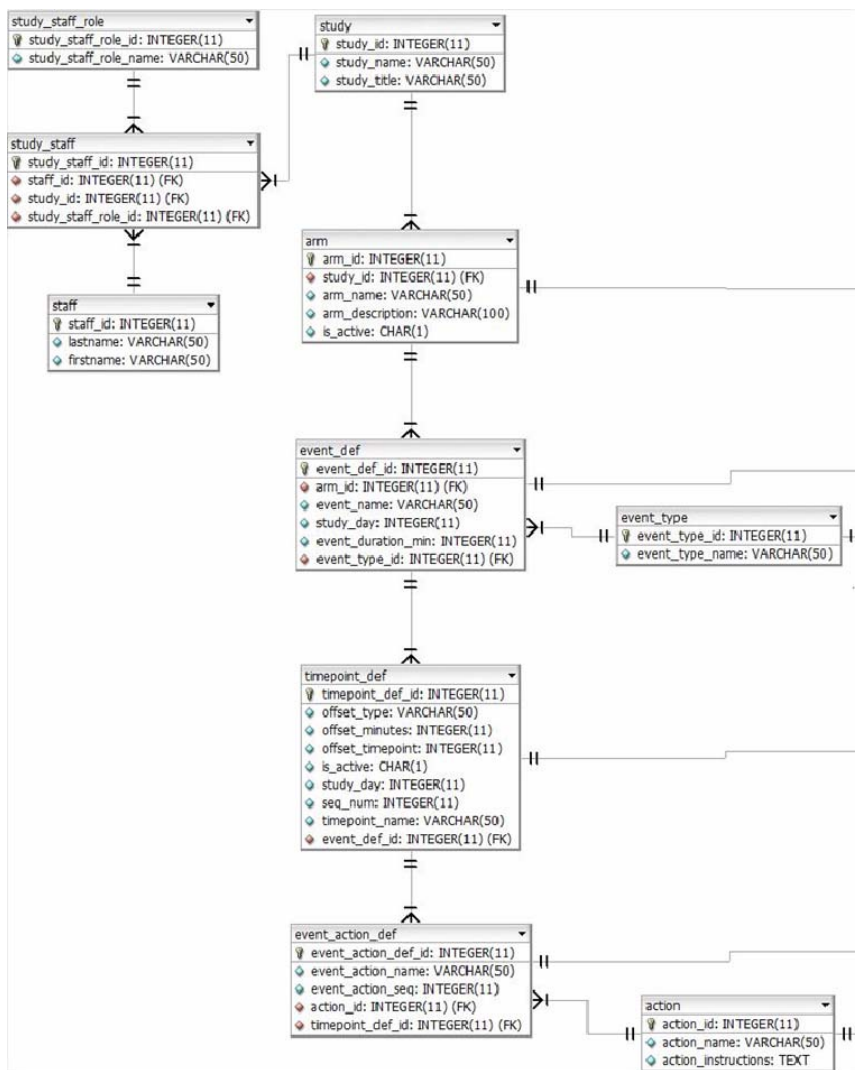


Figure 8: Metadata

Figure 8 represents the Metadata part of the database in its entirety.

The next logical part of our database is the participant part. It starts with a

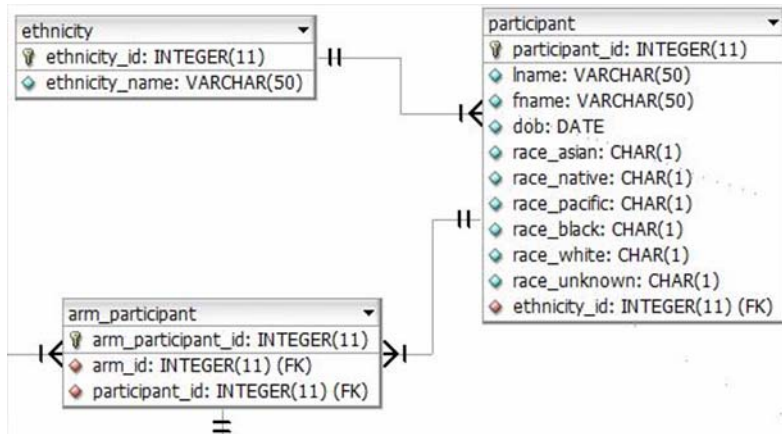


Figure 9: Participant data. Participant and Ethnicity

PARTICIPANT table containing

the basic information about the participant, like his/her name, date of birth, race, and ethnicity. Race represented by Boolean values for each race presented. This is done to accommodate people that associate themselves with more than one race.

ETHNICITY, on the other hand, has a one-to-many relationship to a **PARTICIPANT**, meaning that a participant can only belong to one ethnicity. During enrollment of a participant into the study, **ARM_PARTICIPANT** is the first table that is filled out. We assign a participant to a particular arm of a particular study. After that is done, we can generate a schedule for a participant based on the predetermined rules and data inside our metadata part of the database. Figure 10 shows the structure of the database that accepts the schedule for a participant. This structure is very similar to a metadata structure of an Event-Timepoint-Action relationship with a few additions, in particular timestamps. Since we are dealing with a real participant and real events, timepoints, and actions (as opposed to their definitions on the metadata side of the database) we will need actual times for appointments for each action within each and timepoint under each event therefore we have several timestamps that represent just

that. Each of the events, timepoints, and actions has a status associated with it definitions of which reside in corresponding tables and connected with one-to-many relations.

Translating this structure to our study example we will get a imaginary participant, Stan Smith, to have several outpatient visit scheduled about 6 weeks apart (for each of the "Screening", "Baseline", "Week 6", "Week 12", "Week 18", and so on events). During each outpatient visit Stan will be required to undergo some of the timepoints of the study (Clinical, Cognitive, Thyroid, and so on) each of which is

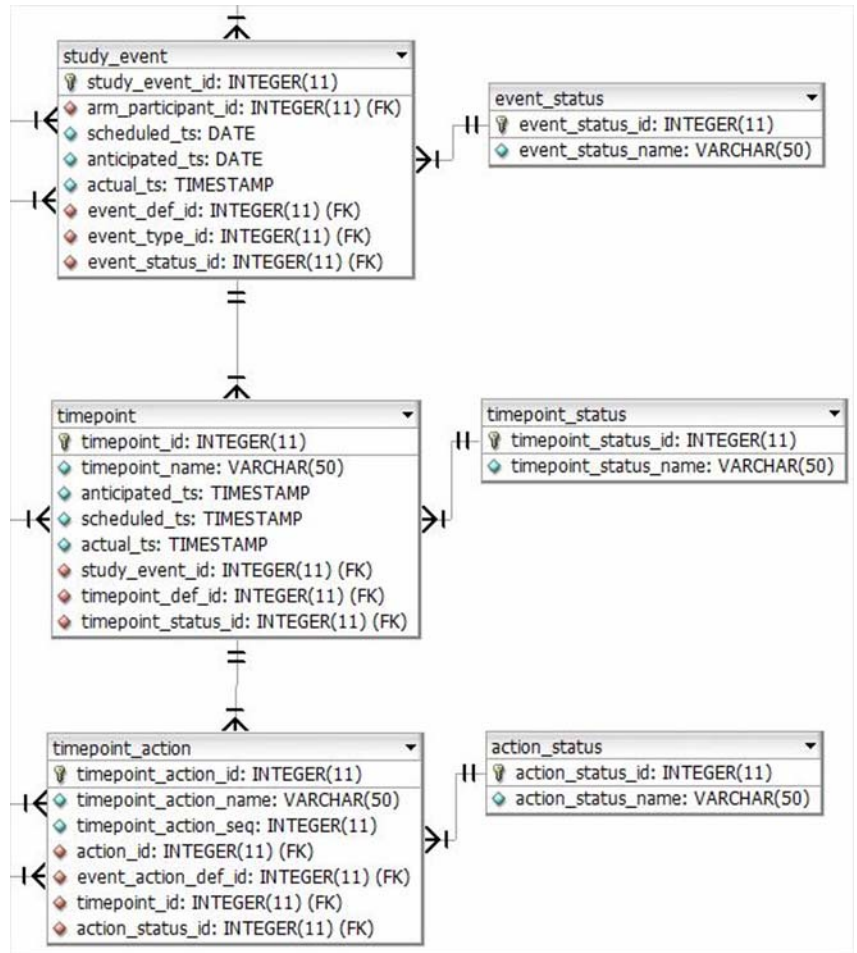


Figure10: Participant schedule

scheduled by the system. During each of the timepoints Stan will be subjected to predetermined tests (actions) like height or weight measurement, TSH level measurement or any other measurement predetermined by the investigator and recorded in the metadata part of the database.

Figure 11 shows the entire database in form of an ER diagram.

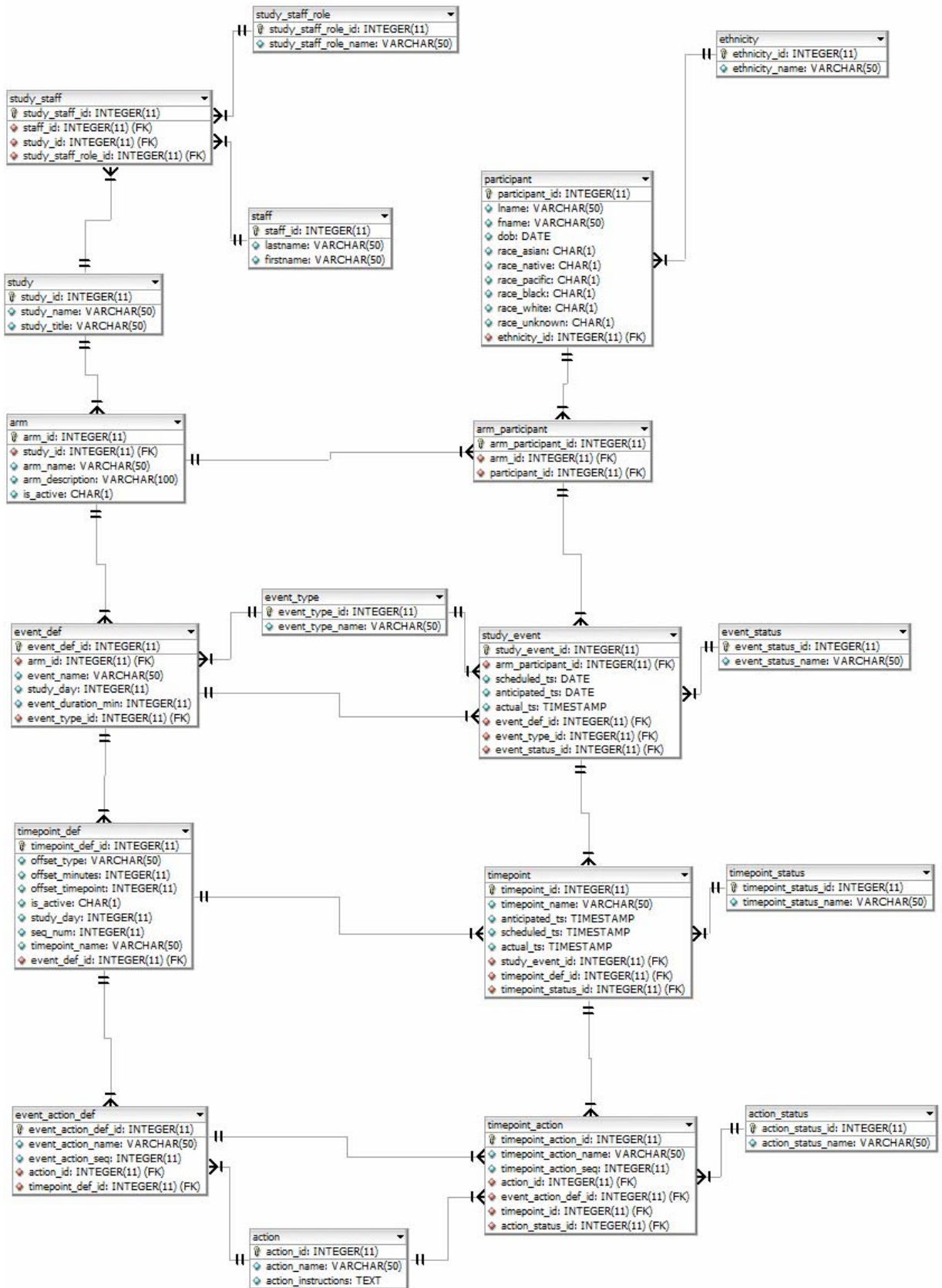


Figure 11: Database

Application

The scope of the application includes creating a secure structure for the web application utilizing the Zend model-viewer-controller framework, connecting the application to the described earlier MySQL database, creating a model-viewer-controller structure for each described table of the database with a Create, Read, Update, and Delete (CRUD) functionalities, and finally implementing a functionality for enrolling a participant into a study and creating a detailed schedule of events for the enrolled participant.

Application structure

One of the useful recommendations of Zend Framework which this project adopted is the directory structure of the web application.

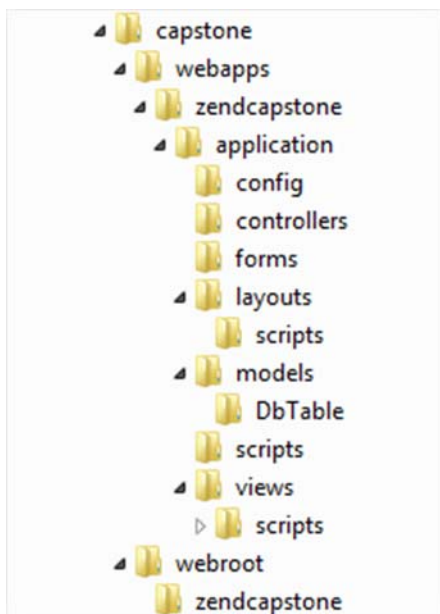


Figure12: Application directory structure

Figure 12 shows the detailed directory structure of the application. The main folder that contains the application is named "capstone". Under it there are two folders named **webapps** and **webroot**. **Webapps** folder contains all of the PHP and HTML code the application. The **webroot** folder is the folder that is public and open to the internet (this folder should be the place your web server points to as a root directory). Both of these folders have the same level in the file system, which means that the actual code for the application will not be

accessible from the web and will be shielded from the potential web attacks. When a request comes from the web to access zendcapstone application, it goes to

`webroot/zendcapstone/index.php` file, which using the rules from `.htaccess` file (resides in the same directory: `webroot/zendcapstone/.htaccess`) redirects it to the `bootstrap.php` file located under `webapps/zendcapstone/application/bootstrap.php`. At this point `bootstrap.php` file sets up the necessary environment for zendcapstone application to execute (including the connection to the database). After everything was successfully set up, the execution is handed down to `webapps/zendcapstone/application/views/scripts/index/index.phtml` which is responsible for displaying the homepage of the application.

All of this happens in the background and the only thing the user needs to do is type the address of the application in the address bar of a web browser.

Figure 13 shows the homepage of the application. The homepage is not impressive because the creation of the graphical user interface was outside the scope of this project.

Recognizable are the names on the homepage

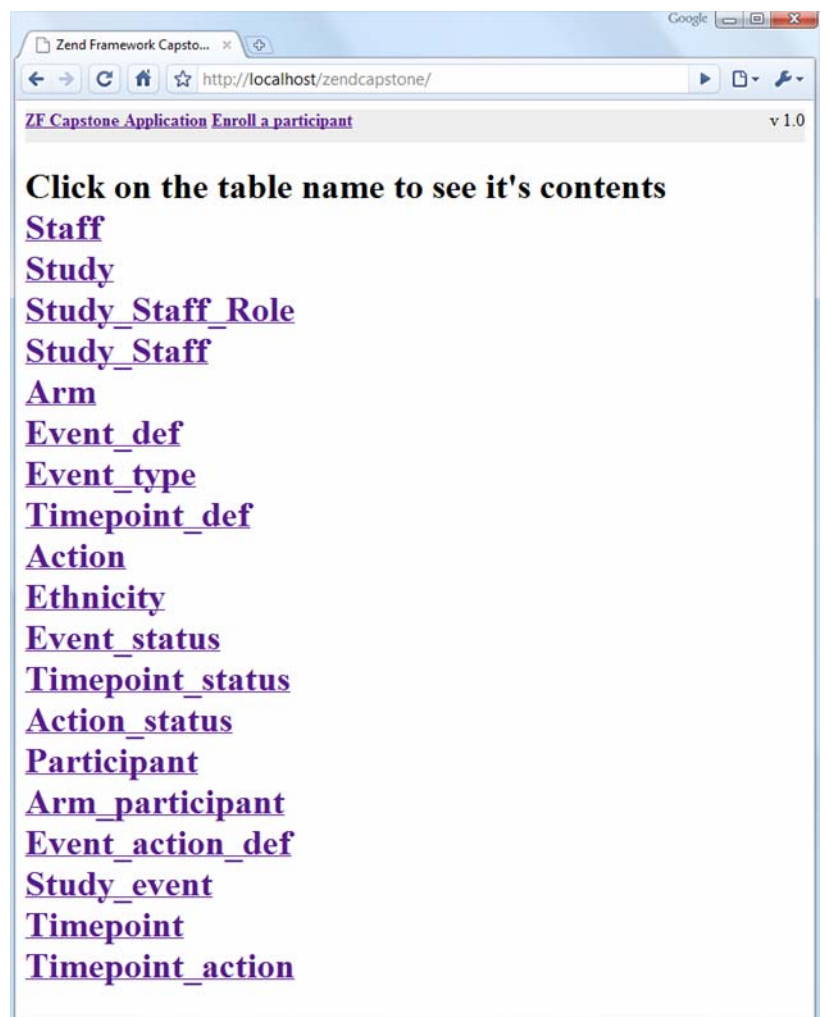


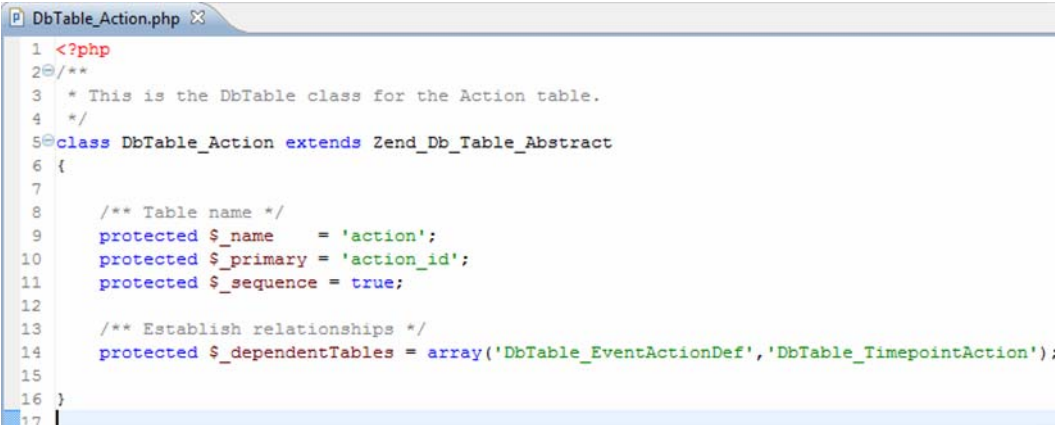
Figure13: Application homepage

because they correspond to the database tables in our database. At the very top of

the page inside a header there are two links. One reads: "ZF Capstone Application" and serves as a link back to the homepage, another one reads: "Enroll a participant" and serves as a function that enrolls a participant into a study and creates a detailed schedule of events for a participant.

Model-View-Controller structure of the application

Each existing table of the database has its own Model-View-Controller structure that is responsible for interacting with the table on the database level, as well as displaying the data from the table to the user. Models for each of the tables and other models used by the application reside inside the `webapps/zendcapstone/application/models` directory. Each model associated with a table has a second level of abstraction that is created to logically divide the functionality and potentially limit the interaction of the code with the database. Let us take a closer look at the structure on the example of one of the tables, we'll choose the **ACTION** table. First, we create our first level of abstraction - the `DbTable_Action.php` file inside `webapps/zendcapstone/application/models/DbTable`



```
1 <?php
2 /**
3  * This is the DbTable class for the Action table.
4  */
5 class DbTable_Action extends Zend_Db_Table_Abstract
6 {
7
8     /** Table name */
9     protected $_name     = 'action';
10    protected $_primary  = 'action_id';
11    protected $_sequence = true;
12
13    /** Establish relationships */
14    protected $_dependentTables = array('DbTable_EventActionDef', 'DbTable_TimepointAction');
15
16 }
17
```

Figure14:DbTable_Action.php

directory. Figure 14 shows the code for this file. Inside of this file we declare `DbTable_Action`

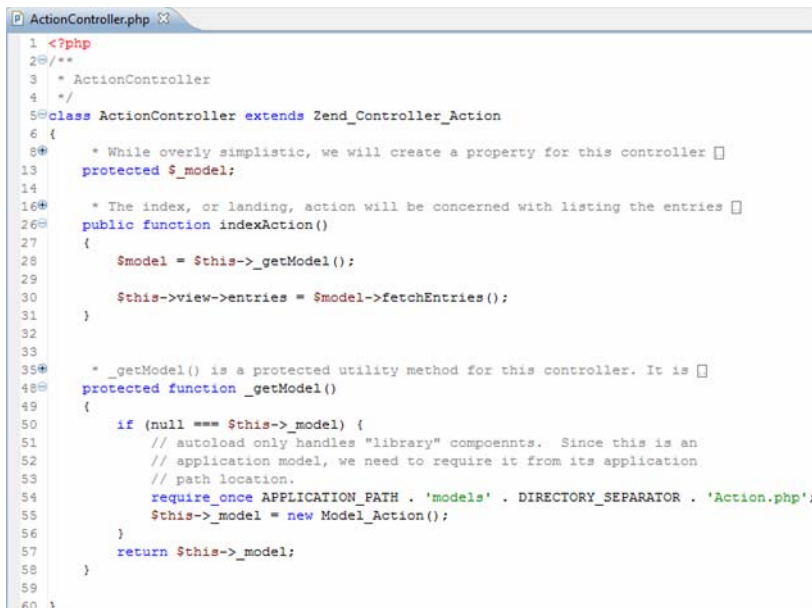
class that extends built-in `Zend_Db_Table_Abstract` class that will be responsible for carrying out all of the functionality for interfacing with the database in whole and the action table in particular. All we need to do is set a few predefined variables like table name and primary key name of the table and Zend code will take care of the rest. The second level of abstraction is the model itself. It comes in form of `Action.php` file that resides in `webapps/zendcapstone/application/models` directory (one level up from our first level of abstraction). Figure 15 contains a snippet of the code.

```
1 <?php
3 * This model class represents the business logic associated with a "Action"
18 class Model_Action
19 {
20     /** Model_Action */
21     protected $_table;
22
23
24 * Retrieve table object
28 public function getTable()
29 {
30     if (null === $this->_table) {
31         // since the dbTable is not a library item but an application item,
32         // we must require it to use it
33         require_once APPLICATION_PATH . 'models' . DIRECTORY_SEPARATOR . 'DbTable' .
34             DIRECTORY_SEPARATOR . 'DbTable_Action.php';
35         $this->_table = new DbTable_Action;
36     }
37     return $this->_table;
38 }
39
40
41 * Function adds a new row into Action table
46 public function insertRow($actionName, $actionInstructions)
47 {
48     //construct an array that contains lastname and firstname
49     $data["action_name"] = $actionName;
50     $data["action_instructions"] = $actionInstructions;
51
52     $table = $this->getTable();
53     return $table->insert($data);
54 }
55
56
57 * Function deletes rows of the action table
61 public function deleteRows(array $ids)
62 {
63     $table = $this->getTable();
64     $deletedRowCount = 0;
65
66     foreach ($ids as $id) {
67         $where = $table->getAdapter()->quoteInto('action_id = ?', $id);
68         $deletedRowCount = $deletedRowCount + $table->delete($where);
69     }
70
71     return $deletedRowCount;
72 }
```

Figure15:Action.php

As soon as we declare the `Model_Action` class we create an instance of the `DbTable_Action` class and use it to interface with the database table. The `Action` model contains code that wraps Zend built-in functions in application-specific functionality. For example, let's take a look at the `insertRow` function. To insert a row into the Action table using Zend built-in functionality the programmer needs to know the exact spelling of each column name of the database and put those names along with the values into a specifically formatted array before Zend Framework can do something useful with it, but by creating a wrapper function inside this model class we can abstract this from the programmer, so the only thing the programmer needs to know to create a new action is the new action's name and action instructions. Each model supports create, read (display), update, and delete functionality using the same abstraction method.

The controller is designed to drive the existing functionality inside the model.



```
1 <?php
2 /**
3  * ActionController
4  */
5 class ActionController extends Zend_Controller_Action
6 {
7
8     * While overly simplistic, we will create a property for this controller []
13    protected $model;
14
15     * The index, or landing, action will be concerned with listing the entries []
16 public function indexAction()
17 {
18     $model = $this->_getModel();
19
20     $this->view->entries = $model->fetchEntries();
21 }
22
23
24     * _getModel() is a protected utility method for this controller. It is []
25 protected function _getModel()
26 {
27     if (null === $this->_model) {
28         // autoload only handles "library" components. Since this is an
29         // application model, we need to require it from its application
30         // path location.
31         require_once APPLICATION_PATH . 'models' . DIRECTORY_SEPARATOR . 'Action.php';
32         $this->_model = new Model_Action();
33     }
34     return $this->_model;
35 }
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60 }
```

Figure16: ActionController.php

Looking at an example with the **ACTION** table we can find a controller for `Model_Action` class inside `webapps/zendcapstone/application/controllers` directory, and according to Zend Framework the filename has to contain a word "Controller" at the end, so in our case we have to name it `ActionController.php`.

The **ActionController** class extends the built-in **Zend_Controller_Action** class (word “Action” is a coincidence; all controllers in our application extend **Zend_Controller_Action** class). This file only contains code for **indexAction** (again, word “Action” is a coincidence) meaning that the **indexAction** function will be executed when the user navigates to **/zendcapstone/Action** or **/zendcapstone/Action/index**. In our case we will show the entire contents of the **ACTION** table, which would look like Figure 17.

The controller for the action table was fairly simple, which would not be the case in the real application. In reality, the controller would contain the code that would respond to user commands to add a row in the table, delete a row, modify or display a row or several rows.

The fact that we can see the entire action table in a nicely formatted table is because we also have a viewer associated with this table. According to our application directory structure all viewers reside inside

action_id	action_name	action_instructions
1	Height	in cm
2	Weight	in kg
3	BP_DIA	in mmHG
4	BP_SYS	in mmHG
5	WBC	count 1000/mm3
6	Hct	%
7	Hgb	gm/dL
8	PLT	count 1000/mm3
9	POMS_T	points, Tension / Anxiety
10	POMS_D	points, Depression / Dejection
11	POMS_A	points, Anger / Hostility
12	POMS_V	points, Vigor
13	POMS_F	points, Fatigue
14	POMS_C	points, Confusion / Bewilderment
15	SOP6	points
16	SOP8	points
17	SOP10	points
18	SOP12	points
19	LDL	mg/dL
20	Trig	mg/dL
21	TSH	mU/L
22	T3	mg/dL
23	FT4	mg/dL

Figure17: Contents of Action table.

`webapps/zendcapstone/application/views/scripts` directory. Because each model-controller structure can have more than one viewer, Zend Framework dictates that we need to create a separate sub-directory for each model-controller structure and place our viewers inside it. Zend Framework is very specific about naming those directories. In our action table example we have to name our sub-directory `action` and the actual file that displays our table must be named `index.phtml` (corresponding to `indexAction` function inside our controller). In other words we have to create the following file that will serve as our viewer: `webapps/zendcapstone/application/views/scripts/action/index.phtml`. Figure 18 shows the contents of that file.

As can be seen, our viewer just loops through an array that was provided by the controller and displays each row inside an HTML table.



```
1 <p>
2
3 Action table entries: <br />
4 <dl>
5
6 <table border="1">
7   <tr>
8     <th>action_id</th>
9     <th>action_name</th>
10    <th>action_instructions</th>
11  </tr>
12
13  <!-- Loop through the entries that were provided to us by the controller -->
14  <? foreach ($this->entries as $entry): ?>
15    <tr><th><?= $this->escape($entry['action_id']) ?></th>
16    <th><?= $this->escape($entry['action_name']) ?></th>
17    <th><?= $this->escape($entry['action_instructions']) ?></th></tr>
18  <? endforeach ?>
19  </table>
20 </dl>
```

Figure18: Viewer for index action

The outlined structure is repeated over and over for each table of the database. Each table has its own `DbTable` class with declared table name, primary key name, and relationship to other tables, model class that knows how to perform CRUD operations, controller class that manipulates the model, and at least one viewer that displays table contents.

Conclusion

The Translational Research Web Application Framework project is an attempt to port existing framework into a newer, more popular technology. It includes a MySQL relational database managing system for a database containing 19 tables, and web application framework written in PHP.

The scope on the project includes:

- Porting of the database from FirebirdSQL to MySQL format.
- Populating the database with an example clinical study
- Creating a directory structure for a web-based application
- Creating a Model-Viewer-Controller structure using Zend Framework
- Creating an easy way for a user to enroll a participant into a clinical study

With this project I have laid a foundation for a PHP web-based application that would be used by clinical study researchers to easily set-up their study, generate participant schedules, and track the progress of the study. Although the application contains working functions to achieve its goal, it still needs work in creating an attractive graphical user interface to make the application usable by a user.

Appendix A. Database creation script

```

/*Create tables */

CREATE TABLE action
(
    action_id            INTEGER NOT NULL AUTO_INCREMENT,
    action_name          VARCHAR(50) NULL,
    action_instructions  TEXT NULL,
    PRIMARY KEY (action_id)
);

CREATE TABLE action_status
(
    action_status_id    INTEGER NOT NULL AUTO_INCREMENT,
    action_status_name  VARCHAR(50) NULL,
    PRIMARY KEY (action_status_id)
);

CREATE TABLE arm
(
    arm_id              INTEGER NOT NULL AUTO_INCREMENT,
    study_id           INTEGER NOT NULL,
    arm_name            VARCHAR(50) NULL,
    arm_description     VARCHAR(100) NULL,
    is_active          CHAR(1) NULL,
    PRIMARY KEY (arm_id)
);

CREATE TABLE arm_participant
(
    arm_participant_id  INTEGER NOT NULL AUTO_INCREMENT,
    arm_id              INTEGER NULL,
    participant_id      INTEGER NOT NULL,
    PRIMARY KEY (arm_participant_id)
);

CREATE TABLE ethnicity
(
    ethnicity_id        INTEGER NOT NULL AUTO_INCREMENT,
    ethnicity_name      VARCHAR(50) NULL,
    PRIMARY KEY (ethnicity_id)
);

CREATE TABLE event_action_def
(
    event_action_def_id INTEGER NOT NULL AUTO_INCREMENT,
    event_action_name   VARCHAR(50) NULL,
    event_action_seq    INTEGER NULL,
    action_id           INTEGER NOT NULL,
    timepoint_def_id   INTEGER NOT NULL,
    PRIMARY KEY (event_action_def_id)
);

CREATE TABLE event_def
(
    event_def_id        INTEGER NOT NULL AUTO_INCREMENT,
    arm_id             INTEGER NOT NULL,
    event_name         VARCHAR(50) NULL,
    study_day          INTEGER NULL,
    event_duration_min INTEGER NULL,
    event_type_id      INTEGER NOT NULL,
    PRIMARY KEY (event_def_id)
);

```

```

CREATE TABLE event_status
(
    event_status_id    INTEGER NOT NULL AUTO_INCREMENT,
    event_status_name  VARCHAR(50) NULL,
    PRIMARY KEY (event_status_id)
);

```

```

CREATE TABLE event_type
(
    event_type_id      INTEGER NOT NULL AUTO_INCREMENT,
    event_type_name    VARCHAR(50) NULL,
    PRIMARY KEY (event_type_id)
);

```

```

CREATE TABLE participant
(
    participant_id     INTEGER NOT NULL AUTO_INCREMENT,
    lname              VARCHAR(50) NULL,
    fname              VARCHAR(50) NULL,
    dob                DATE NULL,
    race_asian         CHAR(1) NULL,
    race_native        CHAR(1) NULL,
    race_pacific       CHAR(1) NULL,
    race_black         CHAR(1) NULL,
    race_white         CHAR(1) NULL,
    race_unknown       CHAR(1) NULL,
    ethnicity_id       INTEGER NULL,
    PRIMARY KEY (participant_id)
);

```

```

CREATE TABLE staff
(
    staff_id           INTEGER NOT NULL AUTO_INCREMENT,
    lastname            VARCHAR(50) NULL,
    firstname           VARCHAR(50) NULL,
    PRIMARY KEY (staff_id)
);

```

```

CREATE TABLE study
(
    study_id           INTEGER NOT NULL AUTO_INCREMENT,
    study_name         VARCHAR(50) NULL,
    study_title        VARCHAR(50) NULL,
    PRIMARY KEY (study_id)
);

```

```

CREATE TABLE study_event
(
    study_event_id     INTEGER NOT NULL AUTO_INCREMENT,
    arm_participant_id INTEGER NULL,
    scheduled_ts        DATE NULL,
    anticipated_ts      DATE NULL,
    actual_ts           TIMESTAMP NULL,
    event_def_id        INTEGER NULL,
    event_type_id       INTEGER NOT NULL,
    event_status_id     INTEGER NOT NULL,
    PRIMARY KEY (study_event_id)
);

```

```

CREATE TABLE study_staff
(
    study_staff_id      INTEGER NOT NULL AUTO_INCREMENT,
    staff_id            INTEGER NOT NULL,
    study_id            INTEGER NOT NULL,
    study_staff_role_id INTEGER NOT NULL,
    PRIMARY KEY (study_staff_id)
);

CREATE TABLE study_staff_role
(
    study_staff_role_id INTEGER NOT NULL AUTO_INCREMENT,
    study_staff_role_name VARCHAR(50) NULL,
    PRIMARY KEY (study_staff_role_id)
);

CREATE TABLE timepoint
(
    timepoint_id        INTEGER NOT NULL AUTO_INCREMENT,
    timepoint_name      VARCHAR(50) NULL,
    anticipated_ts      TIMESTAMP NULL,
    scheduled_ts        TIMESTAMP NULL,
    actual_ts           TIMESTAMP NULL,
    study_event_id      INTEGER NOT NULL,
    timepoint_def_id    INTEGER NULL,
    timepoint_status_id INTEGER NOT NULL,
    PRIMARY KEY (timepoint_id)
);

CREATE TABLE timepoint_action
(
    timepoint_action_id INTEGER NOT NULL AUTO_INCREMENT,
    timepoint_action_name VARCHAR(50) NULL,
    timepoint_action_seq INTEGER NULL,
    action_id           INTEGER NOT NULL,
    event_action_def_id INTEGER NULL,
    timepoint_id        INTEGER NOT NULL,
    action_status_id    INTEGER NOT NULL,
    PRIMARY KEY (timepoint_action_id)
);

CREATE TABLE timepoint_def
(
    timepoint_def_id    INTEGER NOT NULL AUTO_INCREMENT,
    offset_type         VARCHAR(50) NULL,
    offset_minutes      INTEGER NULL,
    offset_timepoint    INTEGER NULL,
    is_active           CHAR(1) NULL,
    study_day           INTEGER NULL,
    seq_num             INTEGER NULL,
    timepoint_name      VARCHAR(50) NULL,
    event_def_id        INTEGER NOT NULL,
    PRIMARY KEY (timepoint_def_id)
);

CREATE TABLE timepoint_status
(
    timepoint_status_id INTEGER NOT NULL AUTO_INCREMENT,
    timepoint_status_name VARCHAR(50) NULL,
    PRIMARY KEY (timepoint_status_id)
);

```

```

/*Create relations between tables*/
ALTER TABLE arm
ADD FOREIGN KEY R_5 (study_id) REFERENCES study (study_id);

ALTER TABLE arm_participant
ADD FOREIGN KEY R_6 (arm_id) REFERENCES arm (arm_id);

ALTER TABLE arm_participant
ADD FOREIGN KEY R_7 (participant_id) REFERENCES participant (participant_id);

ALTER TABLE event_action_def
ADD FOREIGN KEY R_17 (action_id) REFERENCES action (action_id);

ALTER TABLE event_action_def
ADD FOREIGN KEY R_21 (timepoint_def_id) REFERENCES timepoint_def (timepoint_def_id);

ALTER TABLE event_def
ADD FOREIGN KEY R_9 (arm_id) REFERENCES arm (arm_id);

ALTER TABLE event_def
ADD FOREIGN KEY R_14 (event_type_id) REFERENCES event_type (event_type_id);

ALTER TABLE participant
ADD FOREIGN KEY R_25 (ethnicity_id) REFERENCES ethnicity (ethnicity_id);

ALTER TABLE study_event
ADD FOREIGN KEY R_10 (arm_participant_id) REFERENCES arm_participant (arm_participant_id);

ALTER TABLE study_event
ADD FOREIGN KEY R_11 (event_def_id) REFERENCES event_def (event_def_id);

ALTER TABLE study_event
ADD FOREIGN KEY R_13 (event_type_id) REFERENCES event_type (event_type_id);

ALTER TABLE study_event
ADD FOREIGN KEY R_23 (event_status_id) REFERENCES event_status (event_status_id);

ALTER TABLE study_staff
ADD FOREIGN KEY R_3 (staff_id) REFERENCES staff (staff_id);

ALTER TABLE study_staff
ADD FOREIGN KEY R_4 (study_id) REFERENCES study (study_id);

ALTER TABLE study_staff
ADD FOREIGN KEY R_8 (study_staff_role_id) REFERENCES study_staff_role (study_staff_role_id);

ALTER TABLE timepoint
ADD FOREIGN KEY R_12 (study_event_id) REFERENCES study_event (study_event_id);

ALTER TABLE timepoint
ADD FOREIGN KEY R_16 (timepoint_def_id) REFERENCES timepoint_def (timepoint_def_id);

ALTER TABLE timepoint
ADD FOREIGN KEY R_20 (timepoint_status_id) REFERENCES timepoint_status (timepoint_status_id);

ALTER TABLE timepoint_action
ADD FOREIGN KEY R_18 (action_id) REFERENCES action (action_id);

ALTER TABLE timepoint_action
ADD FOREIGN KEY R_19 (event_action_def_id) REFERENCES event_action_def (event_action_def_id);

ALTER TABLE timepoint_action
ADD FOREIGN KEY R_22 (timepoint_id) REFERENCES timepoint (timepoint_id);

ALTER TABLE timepoint_action
ADD FOREIGN KEY R_24 (action_status_id) REFERENCES action_status (action_status_id);

ALTER TABLE timepoint_def
ADD FOREIGN KEY R_15 (event_def_id) REFERENCES event_def (event_def_id);

```

Appendix B. Database population script

```

/*Insert 4 people into staff*/
INSERT INTO staff (lastname,firstname)
VALUES ('Smith','Stan');

INSERT INTO staff (lastname,firstname)
VALUES ('Griffin','Peter');

INSERT INTO staff (lastname,firstname)
VALUES ('Simpson','Bart');

INSERT INTO staff (lastname,firstname)
VALUES ('Lock','John');

/*create 3 study roles*/
INSERT INTO study_staff_role (study_staff_role_name)
VALUES ('Principal Investigator');

INSERT INTO study_staff_role (study_staff_role_name)
VALUES ('Sub-investigator');

INSERT INTO study_staff_role (study_staff_role_name)
VALUES ('Contact');

INSERT INTO study_staff_role (study_staff_role_name)
VALUES ('Coordinator');

/*create study itself*/
INSERT INTO study (study_name,study_title)
VALUES ('Neurocognitive effects of Hypothyroidism ','Effects of Hypothyroidism');

/*populate study_staff with 3 people*/
INSERT INTO study_staff (staff_id,study_id,study_staff_role_id)
VALUES ((SELECT staff_id FROM staff WHERE lastname='Smith'),
        (SELECT study_id FROM study WHERE study_title='Effects of Hypothyroidism'),
        (SELECT study_staff_role_id FROM study_staff_role WHERE study_staff_role_name =
'Principal Investigator'));

INSERT INTO study_staff (staff_id,study_id,study_staff_role_id)
VALUES ((SELECT staff_id FROM staff WHERE lastname='Griffin'),
        (SELECT study_id FROM study WHERE study_title='Effects of Hypothyroidism'),
        (SELECT study_staff_role_id FROM study_staff_role WHERE study_staff_role_name = 'Sub-
investigator'));

INSERT INTO study_staff (staff_id,study_id,study_staff_role_id)
VALUES ((SELECT staff_id FROM staff WHERE lastname='Simpson'),
        (SELECT study_id FROM study WHERE study_title='Effects of Hypothyroidism'),
        (SELECT study_staff_role_id FROM study_staff_role WHERE study_staff_role_name =
'Contact'));

INSERT INTO study_staff (staff_id,study_id,study_staff_role_id)
VALUES ((SELECT staff_id FROM staff WHERE lastname='Lock'),
        (SELECT study_id FROM study WHERE study_title='Effects of Hypothyroidism'),
        (SELECT study_staff_role_id FROM study_staff_role WHERE study_staff_role_name =
'Coordinator'));

/*create arm*/
INSERT INTO arm (study_id,arm_name,arm_description,is_active)
VALUES ((SELECT study_id FROM study WHERE study_title='Effects of Hypothyroidism'),
        'Euthyroid',
        'Treatment method',
        '1');

INSERT INTO arm (study_id,arm_name,arm_description,is_active)
VALUES ((SELECT study_id FROM study WHERE study_title='Effects of Hypothyroidism'),
        'SCH',
        'Treatment method',
        '0');

/*create event type*/
INSERT INTO event_type (event_type_name)
VALUES ('Outpatient visit');

```

```

/*Create events for the study*/
INSERT INTO event_def (arm_id,event_name,study_day,event_duration_min,event_type_id)
VALUES ((SELECT arm_id FROM arm WHERE arm_name='Euthyroid'),
        'Screening',
        0,
        120,
        (SELECT event_type_id FROM event_type WHERE event_type_name='Outpatient visit'));

INSERT INTO event_def (arm_id,event_name,study_day,event_duration_min,event_type_id)
VALUES ((SELECT arm_id FROM arm WHERE arm_name='Euthyroid'),
        'Baseline',
        20,
        150,
        (SELECT event_type_id FROM event_type WHERE event_type_name='Outpatient visit'));

INSERT INTO event_def (arm_id,event_name,study_day,event_duration_min,event_type_id)
VALUES ((SELECT arm_id FROM arm WHERE arm_name='Euthyroid'),
        'Week 6',
        62,
        120,
        (SELECT event_type_id FROM event_type WHERE event_type_name='Outpatient visit'));

INSERT INTO event_def (arm_id,event_name,study_day,event_duration_min,event_type_id)
VALUES ((SELECT arm_id FROM arm WHERE arm_name='Euthyroid'),
        'Week 12',
        104,
        180,
        (SELECT event_type_id FROM event_type WHERE event_type_name='Outpatient visit'));

INSERT INTO event_def (arm_id,event_name,study_day,event_duration_min,event_type_id)
VALUES ((SELECT arm_id FROM arm WHERE arm_name='Euthyroid'),
        'Week 18',
        146,
        150,
        (SELECT event_type_id FROM event_type WHERE event_type_name='Outpatient visit'));

INSERT INTO event_def (arm_id,event_name,study_day,event_duration_min,event_type_id)
VALUES ((SELECT arm_id FROM arm WHERE arm_name='Euthyroid'),
        'Week 24',
        188,
        120,
        (SELECT event_type_id FROM event_type WHERE event_type_name='Outpatient visit'));

INSERT INTO event_def (arm_id,event_name,study_day,event_duration_min,event_type_id)
VALUES ((SELECT arm_id FROM arm WHERE arm_name='Euthyroid'),
        'Week 30',
        230,
        180,
        (SELECT event_type_id FROM event_type WHERE event_type_name='Outpatient visit'));

/*create timepoint, our study will have 6 timepoints per event_def*/
INSERT INTO timepoint_def
(offset_type,offset_minutes,offset_timepoint,is_active,study_day,seq_num,timepoint_name,event_def_id)
VALUES ('event_start',
        0,
        null,
        'Y',
        0,
        1,
        'Clinical',
        (SELECT event_def_id FROM event_def WHERE event_name='Screening'));

INSERT INTO timepoint_def
(offset_type,offset_minutes,offset_timepoint,is_active,study_day,seq_num,timepoint_name,event_def_id)
VALUES ('event_start',
        24,
        null,

```



```

        'Y',
        0,
        2,
        'CBC',
        (SELECT event_def_id FROM event_def WHERE event_name='Screening'));

INSERT INTO timepoint_def
(offset_type,offset_minutes,offset_timepoint,is_active,study_day,seq_num,timepoint_name,event_def_id)
VALUES ('event_start',
        48,
        null,
        'Y',
        0,
        3,
        'Cognitive',
        (SELECT event_def_id FROM event_def WHERE event_name='Screening'));

INSERT INTO timepoint_def
(offset_type,offset_minutes,offset_timepoint,is_active,study_day,seq_num,timepoint_name,event_def_id)
VALUES ('event_start',
        72,
        null,
        'Y',
        0,
        4,
        'Mood',
        (SELECT event_def_id FROM event_def WHERE event_name='Screening'));

INSERT INTO timepoint_def
(offset_type,offset_minutes,offset_timepoint,is_active,study_day,seq_num,timepoint_name,event_def_id)
VALUES ('event_start',
        96,
        null,
        'Y',
        0,
        5,
        'Lipids',
        (SELECT event_def_id FROM event_def WHERE event_name='Screening'));

INSERT INTO timepoint_def
(offset_type,offset_minutes,offset_timepoint,is_active,study_day,seq_num,timepoint_name,event_def_id)
VALUES ('event_start',
        120,
        null,
        'Y',
        0,
        6,
        'Thyroid',
        (SELECT event_def_id FROM event_def WHERE event_name='Screening'));

/* timepoint_defs for baseline*/
INSERT INTO timepoint_def
(offset_type,offset_minutes,offset_timepoint,is_active,study_day,seq_num,timepoint_name,event_def_id)
VALUES ('event_start',
        0,
        null,
        'Y',
        20,
        1,
        'Clinical',

```

```

        (SELECT event_def_id FROM event_def WHERE event_name='Baseline'));

INSERT INTO timepoint_def
(offset_type,offset_minutes,offset_timepoint,is_active,study_day,seq_num,timepoint_name,event_def
_id)
VALUES ('event_start',
        30,
        null,
        'Y',
        20,
        2,
        'CBC',
        (SELECT event_def_id FROM event_def WHERE event_name='Baseline'));

INSERT INTO timepoint_def
(offset_type,offset_minutes,offset_timepoint,is_active,study_day,seq_num,timepoint_name,event_def
_id)
VALUES ('event_start',
        60,
        null,
        'Y',
        20,
        3,
        'Cognitive',
        (SELECT event_def_id FROM event_def WHERE event_name='Baseline'));

INSERT INTO timepoint_def
(offset_type,offset_minutes,offset_timepoint,is_active,study_day,seq_num,timepoint_name,event_def
_id)
VALUES ('event_start',
        90,
        null,
        'Y',
        20,
        4,
        'Mood',
        (SELECT event_def_id FROM event_def WHERE event_name='Baseline'));

INSERT INTO timepoint_def
(offset_type,offset_minutes,offset_timepoint,is_active,study_day,seq_num,timepoint_name,event_def
_id)
VALUES ('event_start',
        120,
        null,
        'Y',
        20,
        5,
        'Lipids',
        (SELECT event_def_id FROM event_def WHERE event_name='Baseline'));

INSERT INTO timepoint_def
(offset_type,offset_minutes,offset_timepoint,is_active,study_day,seq_num,timepoint_name,event_def
_id)
VALUES ('event_start',
        150,
        null,
        'Y',
        20,
        6,
        'Thyroid',
        (SELECT event_def_id FROM event_def WHERE event_name='Baseline'));

/* timepoint_defs for WEEK 6*/
INSERT INTO timepoint_def
(offset_type,offset_minutes,offset_timepoint,is_active,study_day,seq_num,timepoint_name,event_def
_id)

```

```

VALUES ('event_start',
        0,
        null,
        'Y',
        62,
        1,
        'Clinical',
        (SELECT event_def_id FROM event_def WHERE event_name='Week 6'));

INSERT INTO timepoint_def
(offset_type,offset_minutes,offset_timepoint,is_active,study_day,seq_num,timepoint_name,event_def_id)
VALUES ('event_start',
        24,
        null,
        'Y',
        62,
        2,
        'CBC',
        (SELECT event_def_id FROM event_def WHERE event_name='Week 6'));

INSERT INTO timepoint_def
(offset_type,offset_minutes,offset_timepoint,is_active,study_day,seq_num,timepoint_name,event_def_id)
VALUES ('event_start',
        48,
        null,
        'Y',
        62,
        3,
        'Cognitive',
        (SELECT event_def_id FROM event_def WHERE event_name='Week 6'));

INSERT INTO timepoint_def
(offset_type,offset_minutes,offset_timepoint,is_active,study_day,seq_num,timepoint_name,event_def_id)
VALUES ('event_start',
        72,
        null,
        'Y',
        62,
        4,
        'Mood',
        (SELECT event_def_id FROM event_def WHERE event_name='Week 6'));

INSERT INTO timepoint_def
(offset_type,offset_minutes,offset_timepoint,is_active,study_day,seq_num,timepoint_name,event_def_id)
VALUES ('event_start',
        96,
        null,
        'Y',
        62,
        5,
        'Lipids',
        (SELECT event_def_id FROM event_def WHERE event_name='Week 6'));

INSERT INTO timepoint_def
(offset_type,offset_minutes,offset_timepoint,is_active,study_day,seq_num,timepoint_name,event_def_id)
VALUES ('event_start',
        120,
        null,
        'Y',
        62,
        6,
        'Thyroid',

```

```

        (SELECT event_def_id FROM event_def WHERE event_name='Week 6'));

/* timepoint_defs for WEEK 12*/
INSERT INTO timepoint_def
(offset_type,offset_minutes,offset_timepoint,is_active,study_day,seq_num,timepoint_name,event_def
_id)
VALUES ('event_start',
        0,
        null,
        'Y',
        104,
        1,
        'Clinical',
        (SELECT event_def_id FROM event_def WHERE event_name='Week 12'));

INSERT INTO timepoint_def
(offset_type,offset_minutes,offset_timepoint,is_active,study_day,seq_num,timepoint_name,event_def
_id)
VALUES ('event_start',
        36,
        null,
        'Y',
        104,
        2,
        'CBC',
        (SELECT event_def_id FROM event_def WHERE event_name='Week 12'));

INSERT INTO timepoint_def
(offset_type,offset_minutes,offset_timepoint,is_active,study_day,seq_num,timepoint_name,event_def
_id)
VALUES ('event_start',
        72,
        null,
        'Y',
        104,
        3,
        'Cognitive',
        (SELECT event_def_id FROM event_def WHERE event_name='Week 12'));

INSERT INTO timepoint_def
(offset_type,offset_minutes,offset_timepoint,is_active,study_day,seq_num,timepoint_name,event_def
_id)
VALUES ('event_start',
        108,
        null,
        'Y',
        104,
        4,
        'Mood',
        (SELECT event_def_id FROM event_def WHERE event_name='Week 12'));

INSERT INTO timepoint_def
(offset_type,offset_minutes,offset_timepoint,is_active,study_day,seq_num,timepoint_name,event_def
_id)
VALUES ('event_start',
        144,
        null,
        'Y',
        104,
        5,
        'Lipids',
        (SELECT event_def_id FROM event_def WHERE event_name='Week 12'));

INSERT INTO timepoint_def
(offset_type,offset_minutes,offset_timepoint,is_active,study_day,seq_num,timepoint_name,event_def
_id)

```

```

VALUES ('event_start',
        180,
        null,
        'Y',
        104,
        6,
        'Thyroid',
        (SELECT event_def_id FROM event_def WHERE event_name='Week 12'));

/* timepoint_defs for WEEK 18*/
INSERT INTO timepoint_def
(offset_type,offset_minutes,offset_timepoint,is_active,study_day,seq_num,timepoint_name,event_def_id)
VALUES ('event_start',
        0,
        null,
        'Y',
        146,
        1,
        'Clinical',
        (SELECT event_def_id FROM event_def WHERE event_name='Week 18'));

INSERT INTO timepoint_def
(offset_type,offset_minutes,offset_timepoint,is_active,study_day,seq_num,timepoint_name,event_def_id)
VALUES ('event_start',
        30,
        null,
        'Y',
        146,
        2,
        'CBC',
        (SELECT event_def_id FROM event_def WHERE event_name='Week 18'));

INSERT INTO timepoint_def
(offset_type,offset_minutes,offset_timepoint,is_active,study_day,seq_num,timepoint_name,event_def_id)
VALUES ('event_start',
        60,
        null,
        'Y',
        146,
        3,
        'Cognitive',
        (SELECT event_def_id FROM event_def WHERE event_name='Week 18'));

INSERT INTO timepoint_def
(offset_type,offset_minutes,offset_timepoint,is_active,study_day,seq_num,timepoint_name,event_def_id)
VALUES ('event_start',
        90,
        null,
        'Y',
        146,
        4,
        'Mood',
        (SELECT event_def_id FROM event_def WHERE event_name='Week 18'));

INSERT INTO timepoint_def
(offset_type,offset_minutes,offset_timepoint,is_active,study_day,seq_num,timepoint_name,event_def_id)
VALUES ('event_start',
        120,
        null,
        'Y',
        146,
        5,

```

```

        'Lipids',
        (SELECT event_def_id FROM event_def WHERE event_name='Week 18'));

INSERT INTO timepoint_def
(offset_type,offset_minutes,offset_timepoint,is_active,study_day,seq_num,timepoint_name,event_def_id)
VALUES ('event_start',
        150,
        null,
        'Y',
        146,
        6,
        'Thyroid',
        (SELECT event_def_id FROM event_def WHERE event_name='Week 18'));

/* timepoint_defs for WEEK 24*/
INSERT INTO timepoint_def
(offset_type,offset_minutes,offset_timepoint,is_active,study_day,seq_num,timepoint_name,event_def_id)
VALUES ('event_start',
        0,
        null,
        'Y',
        188,
        1,
        'Clinical',
        (SELECT event_def_id FROM event_def WHERE event_name='Week 24'));

INSERT INTO timepoint_def
(offset_type,offset_minutes,offset_timepoint,is_active,study_day,seq_num,timepoint_name,event_def_id)
VALUES ('event_start',
        24,
        null,
        'Y',
        188,
        2,
        'CBC',
        (SELECT event_def_id FROM event_def WHERE event_name='Week 24'));

INSERT INTO timepoint_def
(offset_type,offset_minutes,offset_timepoint,is_active,study_day,seq_num,timepoint_name,event_def_id)
VALUES ('event_start',
        48,
        null,
        'Y',
        188,
        3,
        'Cognitive',
        (SELECT event_def_id FROM event_def WHERE event_name='Week 24'));

INSERT INTO timepoint_def
(offset_type,offset_minutes,offset_timepoint,is_active,study_day,seq_num,timepoint_name,event_def_id)
VALUES ('event_start',
        72,
        null,
        'Y',
        188,
        4,
        'Mood',
        (SELECT event_def_id FROM event_def WHERE event_name='Week 24'));

INSERT INTO timepoint_def
(offset_type,offset_minutes,offset_timepoint,is_active,study_day,seq_num,timepoint_name,event_def_id)

```

```

VALUES ('event_start',
        96,
        null,
        'Y',
        188,
        5,
        'Lipids',
        (SELECT event_def_id FROM event_def WHERE event_name='Week 24'));

INSERT INTO timepoint_def
(offset_type,offset_minutes,offset_timepoint,is_active,study_day,seq_num,timepoint_name,event_def_id)
VALUES ('event_start',
        120,
        null,
        'Y',
        188,
        6,
        'Thyroid',
        (SELECT event_def_id FROM event_def WHERE event_name='Week 24'));

/* timepoint_defs for WEEK 30*/
INSERT INTO timepoint_def
(offset_type,offset_minutes,offset_timepoint,is_active,study_day,seq_num,timepoint_name,event_def_id)
VALUES ('event_start',
        0,
        null,
        'Y',
        230,
        1,
        'Clinical',
        (SELECT event_def_id FROM event_def WHERE event_name='Week 30'));

INSERT INTO timepoint_def
(offset_type,offset_minutes,offset_timepoint,is_active,study_day,seq_num,timepoint_name,event_def_id)
VALUES ('event_start',
        36,
        null,
        'Y',
        230,
        2,
        'CBC',
        (SELECT event_def_id FROM event_def WHERE event_name='Week 30'));

INSERT INTO timepoint_def
(offset_type,offset_minutes,offset_timepoint,is_active,study_day,seq_num,timepoint_name,event_def_id)
VALUES ('event_start',
        72,
        null,
        'Y',
        230,
        3,
        'Cognitive',
        (SELECT event_def_id FROM event_def WHERE event_name='Week 30'));

INSERT INTO timepoint_def
(offset_type,offset_minutes,offset_timepoint,is_active,study_day,seq_num,timepoint_name,event_def_id)
VALUES ('event_start',
        108,
        null,
        'Y',
        230,
        4,

```

```

        'Mood',
        (SELECT event_def_id FROM event_def WHERE event_name='Week 30'));

INSERT INTO timepoint_def
(offset_type,offset_minutes,offset_timepoint,is_active,study_day,seq_num,timepoint_name,event_def_id)
VALUES ('event_start',
        144,
        null,
        'Y',
        230,
        5,
        'Lipids',
        (SELECT event_def_id FROM event_def WHERE event_name='Week 30'));

INSERT INTO timepoint_def
(offset_type,offset_minutes,offset_timepoint,is_active,study_day,seq_num,timepoint_name,event_def_id)
VALUES ('event_start',
        180,
        null,
        'Y',
        230,
        6,
        'Thyroid',
        (SELECT event_def_id FROM event_def WHERE event_name='Week 30'));

/*now create actions*/
INSERT INTO action (action_name, action_instructions)
VALUES ('Height', 'in cm');

INSERT INTO action (action_name, action_instructions)
VALUES ('Weight','in kg');

INSERT INTO action (action_name, action_instructions)
VALUES ('BP_DIA','in mmHG');

INSERT INTO action (action_name, action_instructions)
VALUES ('BP_SYS','in mmHG');

INSERT INTO action (action_name, action_instructions)
VALUES ('WBC','count 1000/mm3');

INSERT INTO action (action_name, action_instructions)
VALUES ('Hct','%');

INSERT INTO action (action_name, action_instructions)
VALUES ('Hgb','gm/dL');

INSERT INTO action (action_name, action_instructions)
VALUES ('PLT','count 1000/mm3');

INSERT INTO action (action_name, action_instructions)
VALUES ('POMS_T','points, Tension / Anxiety');

INSERT INTO action (action_name, action_instructions)
VALUES ('POMS_D','points, Depression / Dejection');

INSERT INTO action (action_name, action_instructions)
VALUES ('POMS_A','points, Anger / Hostility');

INSERT INTO action (action_name, action_instructions)
VALUES ('POMS_V','points, Vigor');

INSERT INTO action (action_name, action_instructions)
VALUES ('POMS_F','points, Fatigue');

INSERT INTO action (action_name, action_instructions)

```



```

VALUES ('POMS_C','points, Confusion / Bewilderment');

INSERT INTO action (action_name, action_instructions)
VALUES ('SOP6','points');

INSERT INTO action (action_name, action_instructions)
VALUES ('SOP8','points');

INSERT INTO action (action_name, action_instructions)
VALUES ('SOP10','points');

INSERT INTO action (action_name, action_instructions)
VALUES ('SOP12','points');

INSERT INTO action (action_name, action_instructions)
VALUES ('LDL','mg/dL');

INSERT INTO action (action_name, action_instructions)
VALUES ('Trig','mg/dL');

INSERT INTO action (action_name, action_instructions)
VALUES ('TSH','mU/L');

INSERT INTO action (action_name, action_instructions)
VALUES ('T3','mg/dL');

INSERT INTO action (action_name, action_instructions)
VALUES ('FT4','mg/dL');

/*now populate event_action_def, assign actions to Screening, Clinical timepoint*/
INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('Initial Height',
      1,
      (SELECT action_id FROM action WHERE action_name = 'Height'),
      (SELECT timepoint_def_id
        FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
        WHERE t.timepoint_name = 'Clinical' and e.event_name = 'Screening'));

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('Initial Weight',
      2,
      (SELECT action_id FROM action WHERE action_name = 'Weight'),
      (SELECT timepoint_def_id
        FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
        WHERE t.timepoint_name = 'Clinical' and e.event_name = 'Screening'));

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('Initial BP_SYS',
      3,
      (SELECT action_id FROM action WHERE action_name = 'BP_SYS'),
      (SELECT timepoint_def_id
        FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
        WHERE t.timepoint_name = 'Clinical' and e.event_name = 'Screening'));

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('Initial BP_DIA',
      4,
      (SELECT action_id FROM action WHERE action_name = 'BP_DIA'),
      (SELECT timepoint_def_id
        FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
        WHERE t.timepoint_name = 'Clinical' and e.event_name = 'Screening'));

/*now populate event_action_def, assign actions to Screening, CBC timepoint*/
INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('Initial WBC',

```

```

1,
  (SELECT action_id FROM action WHERE action_name = 'WBC'),
  (SELECT timepoint_def_id
    FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
    WHERE t.timepoint_name = 'CBC' and e.event_name = 'Screening'));

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('Initial Hc',
  2,
  (SELECT action_id FROM action WHERE action_name = 'Hct'),
  (SELECT timepoint_def_id
    FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
    WHERE t.timepoint_name = 'CBC' and e.event_name = 'Screening'));

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('Initial Hg',
  3,
  (SELECT action_id FROM action WHERE action_name = 'Hgb'),
  (SELECT timepoint_def_id
    FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
    WHERE t.timepoint_name = 'CBC' and e.event_name = 'Screening'));

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('Initial Platelet',
  4,
  (SELECT action_id FROM action WHERE action_name = 'PLT'),
  (SELECT timepoint_def_id
    FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
    WHERE t.timepoint_name = 'CBC' and e.event_name = 'Screening'));

/*now populate event_action_def, assign actions to Screening, LIPIDS timepoint*/
INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('Initial LDL',
  1,
  (SELECT action_id FROM action WHERE action_name = 'LDL'),
  (SELECT timepoint_def_id
    FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
    WHERE t.timepoint_name = 'Lipids' and e.event_name = 'Screening'));

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('Initial Trig',
  2,
  (SELECT action_id FROM action WHERE action_name = 'Trig'),
  (SELECT timepoint_def_id
    FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
    WHERE t.timepoint_name = 'Lipids' and e.event_name = 'Screening'));

/*now populate event_action_def, assign actions to Screening, THYROID timepoint*/
INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('Initial TSH',
  1,
  (SELECT action_id FROM action WHERE action_name = 'TSH'),
  (SELECT timepoint_def_id
    FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
    WHERE t.timepoint_name = 'Thyroid' and e.event_name = 'Screening'));

/*now populate event_action_def, assign actions to BASELINE, CLINICAL timepoint*/
INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('Weight',
  1,

```

```

        (SELECT action_id FROM action WHERE action_name = 'Weight'),
        (SELECT timepoint_def_id
         FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
         WHERE t.timepoint_name = 'Clinical' and e.event_name = 'Baseline'));

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('BP_SYS',
        2,
        (SELECT action_id FROM action WHERE action_name = 'BP_SYS'),
        (SELECT timepoint_def_id
         FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
         WHERE t.timepoint_name = 'Clinical' and e.event_name = 'Baseline'));

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('BP_DIA',
        3,
        (SELECT action_id FROM action WHERE action_name = 'BP_DIA'),
        (SELECT timepoint_def_id
         FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
         WHERE t.timepoint_name = 'Clinical' and e.event_name = 'Baseline'));

/*now populate event_action_def, assign actions to BASELINE, COGNITIVE timepoint*/
INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('SOP6',
        1,
        (SELECT action_id FROM action WHERE action_name = 'SOP6'),
        (SELECT timepoint_def_id
         FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
         WHERE t.timepoint_name = 'Cognitive' and e.event_name = 'Baseline'));

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('SOP8',
        2,
        (SELECT action_id FROM action WHERE action_name = 'SOP8'),
        (SELECT timepoint_def_id
         FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
         WHERE t.timepoint_name = 'Cognitive' and e.event_name = 'Baseline'));

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('SOP10',
        3,
        (SELECT action_id FROM action WHERE action_name = 'SOP10'),
        (SELECT timepoint_def_id
         FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
         WHERE t.timepoint_name = 'Cognitive' and e.event_name = 'Baseline'));

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('SOP12',
        4,
        (SELECT action_id FROM action WHERE action_name = 'SOP12'),
        (SELECT timepoint_def_id
         FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
         WHERE t.timepoint_name = 'Cognitive' and e.event_name = 'Baseline'));

/*now populate event_action_def, assign actions to BASELINE, MOOD timepoint*/
INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('POMSA',
        1,
        (SELECT action_id FROM action WHERE action_name = 'POMS_A'),
        (SELECT timepoint_def_id
         FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id

```

```

WHERE t.timepoint_name = 'Mood' and e.event_name = 'Baseline'));

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('POMSC',
        2,
        (SELECT action_id FROM action WHERE action_name = 'POMS_C'),
        (SELECT timepoint_def_id
         FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
         WHERE t.timepoint_name = 'Mood' and e.event_name = 'Baseline'));

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('POMST',
        3,
        (SELECT action_id FROM action WHERE action_name = 'POMS_T'),
        (SELECT timepoint_def_id
         FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
         WHERE t.timepoint_name = 'Mood' and e.event_name = 'Baseline'));

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('POMSD',
        4,
        (SELECT action_id FROM action WHERE action_name = 'POMS_D'),
        (SELECT timepoint_def_id
         FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
         WHERE t.timepoint_name = 'Mood' and e.event_name = 'Baseline'));

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('POMSF',
        5,
        (SELECT action_id FROM action WHERE action_name = 'POMS_F'),
        (SELECT timepoint_def_id
         FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
         WHERE t.timepoint_name = 'Mood' and e.event_name = 'Baseline'));

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('POMSV',
        6,
        (SELECT action_id FROM action WHERE action_name = 'POMS_V'),
        (SELECT timepoint_def_id
         FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
         WHERE t.timepoint_name = 'Mood' and e.event_name = 'Baseline'));

/*now populate event_action_def, assign actions to BASELINE, THYROID timepoint*/
INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('TSH',
        1,
        (SELECT action_id FROM action WHERE action_name = 'TSH'),
        (SELECT timepoint_def_id
         FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
         WHERE t.timepoint_name = 'Thyroid' and e.event_name = 'Baseline'));

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('T3',
        2,
        (SELECT action_id FROM action WHERE action_name = 'T3'),
        (SELECT timepoint_def_id
         FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
         WHERE t.timepoint_name = 'Thyroid' and e.event_name = 'Baseline'));

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('FT4',
        3,
        (SELECT action_id FROM action WHERE action_name = 'FT4'),

```

```

        (SELECT timepoint_def_id
         FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
         WHERE t.timepoint_name = 'Thyroid' and e.event_name = 'Baseline'));

/*now populate event_action_def, assign actions to WEEK 6, CLINICAL timepoint*/
INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('Weight',
        1,
        (SELECT action_id FROM action WHERE action_name = 'Weight'),
        (SELECT timepoint_def_id
         FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
         WHERE t.timepoint_name = 'Clinical' and e.event_name = 'Week 6'));

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('BP_SYS',
        2,
        (SELECT action_id FROM action WHERE action_name = 'BP_SYS'),
        (SELECT timepoint_def_id
         FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
         WHERE t.timepoint_name = 'Clinical' and e.event_name = 'Week 6'));

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('BP_DIA',
        3,
        (SELECT action_id FROM action WHERE action_name = 'BP_DIA'),
        (SELECT timepoint_def_id
         FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
         WHERE t.timepoint_name = 'Clinical' and e.event_name = 'Week 6'));

/*now populate event_action_def, assign actions to WEEK 6, LIPIDS timepoint*/
INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('LDL',
        1,
        (SELECT action_id FROM action WHERE action_name = 'LDL'),
        (SELECT timepoint_def_id
         FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
         WHERE t.timepoint_name = 'Lipids' and e.event_name = 'Week 6'));

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('Trig',
        2,
        (SELECT action_id FROM action WHERE action_name = 'Trig'),
        (SELECT timepoint_def_id
         FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
         WHERE t.timepoint_name = 'Lipids' and e.event_name = 'Week 6'));

/*now populate event_action_def, assign actions to Week 6, THYROID timepoint*/
INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('TSH',
        1,
        (SELECT action_id FROM action WHERE action_name = 'TSH'),
        (SELECT timepoint_def_id
         FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
         WHERE t.timepoint_name = 'Thyroid' and e.event_name = 'Week 6'));

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('T3',
        2,
        (SELECT action_id FROM action WHERE action_name = 'T3'),
        (SELECT timepoint_def_id

```

```

        FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
        WHERE t.timepoint_name = 'Thyroid' and e.event_name = 'Week 6'));

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('FT4',
        3,
        (SELECT action_id FROM action WHERE action_name = 'FT4'),
        (SELECT timepoint_def_id
        FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
        WHERE t.timepoint_name = 'Thyroid' and e.event_name = 'Week 6')));

/*now populate event_action_def, assign actions to WEEK 12, CLINICAL timepoint*/
INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('Weight',
        1,
        (SELECT action_id FROM action WHERE action_name = 'Weight'),
        (SELECT timepoint_def_id
        FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
        WHERE t.timepoint_name = 'Clinical' and e.event_name = 'Week 12'));

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('BP_SYS',
        2,
        (SELECT action_id FROM action WHERE action_name = 'BP_SYS'),
        (SELECT timepoint_def_id
        FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
        WHERE t.timepoint_name = 'Clinical' and e.event_name = 'Week 12'));

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('BP_DIA',
        3,
        (SELECT action_id FROM action WHERE action_name = 'BP_DIA'),
        (SELECT timepoint_def_id
        FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
        WHERE t.timepoint_name = 'Clinical' and e.event_name = 'Week 12'));

/*now populate event_action_def, assign actions to Week 12, COGNITIVE timepoint*/
INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('SOP6',
        1,
        (SELECT action_id FROM action WHERE action_name = 'SOP6'),
        (SELECT timepoint_def_id
        FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
        WHERE t.timepoint_name = 'Cognitive' and e.event_name = 'Week 12'));

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('SOP8',
        2,
        (SELECT action_id FROM action WHERE action_name = 'SOP8'),
        (SELECT timepoint_def_id
        FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
        WHERE t.timepoint_name = 'Cognitive' and e.event_name = 'Week 12'));

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('SOP10',
        3,
        (SELECT action_id FROM action WHERE action_name = 'SOP10'),
        (SELECT timepoint_def_id
        FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
        WHERE t.timepoint_name = 'Cognitive' and e.event_name = 'Week 12'));

```

```

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('SOP12',
      4,
      (SELECT action_id FROM action WHERE action_name = 'SOP12'),
      (SELECT timepoint_def_id
        FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
        WHERE t.timepoint_name = 'Cognitive' and e.event_name = 'Week 12'));

/*now populate event_action_def, assign actions to BASELINE, MOOD timepoint*/
INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('POMSA',
      1,
      (SELECT action_id FROM action WHERE action_name = 'POMS_A'),
      (SELECT timepoint_def_id
        FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
        WHERE t.timepoint_name = 'Mood' and e.event_name = 'Week 12'));

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('POMSC',
      2,
      (SELECT action_id FROM action WHERE action_name = 'POMS_C'),
      (SELECT timepoint_def_id
        FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
        WHERE t.timepoint_name = 'Mood' and e.event_name = 'Week 12'));

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('POMST',
      3,
      (SELECT action_id FROM action WHERE action_name = 'POMS_T'),
      (SELECT timepoint_def_id
        FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
        WHERE t.timepoint_name = 'Mood' and e.event_name = 'Week 12'));

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('POMSD',
      4,
      (SELECT action_id FROM action WHERE action_name = 'POMS_D'),
      (SELECT timepoint_def_id
        FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
        WHERE t.timepoint_name = 'Mood' and e.event_name = 'Week 12'));

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('POMSF',
      5,
      (SELECT action_id FROM action WHERE action_name = 'POMS_F'),
      (SELECT timepoint_def_id
        FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
        WHERE t.timepoint_name = 'Mood' and e.event_name = 'Week 12'));

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('POMSV',
      6,
      (SELECT action_id FROM action WHERE action_name = 'POMS_V'),
      (SELECT timepoint_def_id
        FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
        WHERE t.timepoint_name = 'Mood' and e.event_name = 'Week 12'));

/*now populate event_action_def, assign actions to Week 12, THYROID timepoint*/
INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('TSH',
      1,

```

```

        (SELECT action_id FROM action WHERE action_name = 'TSH'),
        (SELECT timepoint_def_id
         FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
         WHERE t.timepoint_name = 'Thyroid' and e.event_name = 'Week 12'));

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('T3',
        2,
        (SELECT action_id FROM action WHERE action_name = 'T3'),
        (SELECT timepoint_def_id
         FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
         WHERE t.timepoint_name = 'Thyroid' and e.event_name = 'Week 12'));

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('FT4',
        3,
        (SELECT action_id FROM action WHERE action_name = 'FT4'),
        (SELECT timepoint_def_id
         FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
         WHERE t.timepoint_name = 'Thyroid' and e.event_name = 'Week 12'));

/*now populate event_action_def, assign actions to WEEK 18, CLINICAL timepoint*/
INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('Weight',
        1,
        (SELECT action_id FROM action WHERE action_name = 'Weight'),
        (SELECT timepoint_def_id
         FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
         WHERE t.timepoint_name = 'Clinical' and e.event_name = 'Week 18'));

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('BP_SYS',
        2,
        (SELECT action_id FROM action WHERE action_name = 'BP_SYS'),
        (SELECT timepoint_def_id
         FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
         WHERE t.timepoint_name = 'Clinical' and e.event_name = 'Week 18'));

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('BP_DIA',
        3,
        (SELECT action_id FROM action WHERE action_name = 'BP_DIA'),
        (SELECT timepoint_def_id
         FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
         WHERE t.timepoint_name = 'Clinical' and e.event_name = 'Week 18'));

/*now populate event_action_def, assign actions to WEEK 18, LIPIDS timepoint*/
INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('LDL',
        1,
        (SELECT action_id FROM action WHERE action_name = 'LDL'),
        (SELECT timepoint_def_id
         FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
         WHERE t.timepoint_name = 'Lipids' and e.event_name = 'Week 18'));

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('Trig',
        2,
        (SELECT action_id FROM action WHERE action_name = 'Trig'),
        (SELECT timepoint_def_id
         FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id

```



```

WHERE t.timepoint_name = 'Lipids' and e.event_name = 'Week 18'));

/*now populate event_action_def, assign actions to Week 18, THYROID timepoint*/
INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('TSH',
        1,
        (SELECT action_id FROM action WHERE action_name = 'TSH'),
        (SELECT timepoint_def_id
         FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
         WHERE t.timepoint_name = 'Thyroid' and e.event_name = 'Week 18'));

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('T3',
        2,
        (SELECT action_id FROM action WHERE action_name = 'T3'),
        (SELECT timepoint_def_id
         FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
         WHERE t.timepoint_name = 'Thyroid' and e.event_name = 'Week 18'));

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('FT4',
        3,
        (SELECT action_id FROM action WHERE action_name = 'FT4'),
        (SELECT timepoint_def_id
         FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
         WHERE t.timepoint_name = 'Thyroid' and e.event_name = 'Week 18'));

/*now populate event_action_def, assign actions to WEEK 24, CLINICAL timepoint*/
INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('Weight',
        1,
        (SELECT action_id FROM action WHERE action_name = 'Weight'),
        (SELECT timepoint_def_id
         FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
         WHERE t.timepoint_name = 'Clinical' and e.event_name = 'Week 24'));

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('BP_SYS',
        2,
        (SELECT action_id FROM action WHERE action_name = 'BP_SYS'),
        (SELECT timepoint_def_id
         FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
         WHERE t.timepoint_name = 'Clinical' and e.event_name = 'Week 24'));

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('BP_DIA',
        3,
        (SELECT action_id FROM action WHERE action_name = 'BP_DIA'),
        (SELECT timepoint_def_id
         FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
         WHERE t.timepoint_name = 'Clinical' and e.event_name = 'Week 24'));

/*now populate event_action_def, assign actions to Week 24, COGNITIVE timepoint*/
INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('SOP6',
        1,
        (SELECT action_id FROM action WHERE action_name = 'SOP6'),
        (SELECT timepoint_def_id
         FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
         WHERE t.timepoint_name = 'Cognitive' and e.event_name = 'Week 24'));

```

```

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('SOP8',
        2,
        (SELECT action_id FROM action WHERE action_name = 'SOP8'),
        (SELECT timepoint_def_id
         FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
         WHERE t.timepoint_name = 'Cognitive' and e.event_name = 'Week 24'));

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('SOP10',
        3,
        (SELECT action_id FROM action WHERE action_name = 'SOP10'),
        (SELECT timepoint_def_id
         FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
         WHERE t.timepoint_name = 'Cognitive' and e.event_name = 'Week 24'));

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('SOP12',
        4,
        (SELECT action_id FROM action WHERE action_name = 'SOP12'),
        (SELECT timepoint_def_id
         FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
         WHERE t.timepoint_name = 'Cognitive' and e.event_name = 'Week 24'));

/*now populate event_action_def, assign actions to BASELINE, MOOD timepoint*/
INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('POMSA',
        1,
        (SELECT action_id FROM action WHERE action_name = 'POMS_A'),
        (SELECT timepoint_def_id
         FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
         WHERE t.timepoint_name = 'Mood' and e.event_name = 'Week 24'));

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('POMSC',
        2,
        (SELECT action_id FROM action WHERE action_name = 'POMS_C'),
        (SELECT timepoint_def_id
         FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
         WHERE t.timepoint_name = 'Mood' and e.event_name = 'Week 24'));

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('POMST',
        3,
        (SELECT action_id FROM action WHERE action_name = 'POMS_T'),
        (SELECT timepoint_def_id
         FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
         WHERE t.timepoint_name = 'Mood' and e.event_name = 'Week 24'));

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('POMSD',
        4,
        (SELECT action_id FROM action WHERE action_name = 'POMS_D'),
        (SELECT timepoint_def_id
         FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
         WHERE t.timepoint_name = 'Mood' and e.event_name = 'Week 24'));

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('POMSF',
        5,
        (SELECT action_id FROM action WHERE action_name = 'POMS_F'),
        (SELECT timepoint_def_id

```

```

        FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
        WHERE t.timepoint_name = 'Mood' and e.event_name = 'Week 24'));

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('POMSV',
        6,
        (SELECT action_id FROM action WHERE action_name = 'POMS_V'),
        (SELECT timepoint_def_id
        FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
        WHERE t.timepoint_name = 'Mood' and e.event_name = 'Week 24'));

/*now populate event_action_def, assign actions to Week 24, THYROID timepoint*/
INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('TSH',
        1,
        (SELECT action_id FROM action WHERE action_name = 'TSH'),
        (SELECT timepoint_def_id
        FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
        WHERE t.timepoint_name = 'Thyroid' and e.event_name = 'Week 24'));

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('T3',
        2,
        (SELECT action_id FROM action WHERE action_name = 'T3'),
        (SELECT timepoint_def_id
        FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
        WHERE t.timepoint_name = 'Thyroid' and e.event_name = 'Week 24'));

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('FT4',
        3,
        (SELECT action_id FROM action WHERE action_name = 'FT4'),
        (SELECT timepoint_def_id
        FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
        WHERE t.timepoint_name = 'Thyroid' and e.event_name = 'Week 24'));

/*now populate event_action_def, assign actions to WEEK 30, CLINICAL timepoint*/
INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('Weight',
        1,
        (SELECT action_id FROM action WHERE action_name = 'Weight'),
        (SELECT timepoint_def_id
        FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
        WHERE t.timepoint_name = 'Clinical' and e.event_name = 'Week 30'));

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('BP_SYS',
        2,
        (SELECT action_id FROM action WHERE action_name = 'BP_SYS'),
        (SELECT timepoint_def_id
        FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
        WHERE t.timepoint_name = 'Clinical' and e.event_name = 'Week 30'));

INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('BP_DIA',
        3,
        (SELECT action_id FROM action WHERE action_name = 'BP_DIA'),
        (SELECT timepoint_def_id
        FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
        WHERE t.timepoint_name = 'Clinical' and e.event_name = 'Week 30'));

```

```

/*now populate event_action_def, assign actions to WEEK 30, Thyroid timepoint*/
INSERT INTO event_action_def (event_action_name, event_action_seq, action_id, timepoint_def_id)
VALUES ('TSH',
      1,
      (SELECT action_id FROM action WHERE action_name = 'TSH'),
      (SELECT timepoint_def_id
        FROM timepoint_def t INNER JOIN event_def e ON t.event_def_id =
e.event_def_id
        WHERE t.timepoint_name = 'Thyroid' and e.event_name = 'Week 30'));

/*populate ethnicity*/
INSERT INTO ethnicity (ethnicity_name)
VALUES ('Hispanic');

INSERT INTO ethnicity (ethnicity_name)
VALUES ('Native American');

INSERT INTO ethnicity (ethnicity_name)
VALUES ('Arab');

INSERT INTO ethnicity (ethnicity_name)
VALUES ('African');

INSERT INTO ethnicity (ethnicity_name)
VALUES ('Hindu');

INSERT INTO ethnicity (ethnicity_name)
VALUES ('Slavic');

INSERT INTO ethnicity (ethnicity_name)
VALUES ('White');

INSERT INTO ethnicity (ethnicity_name)
VALUES ('Orient');

INSERT INTO ethnicity (ethnicity_name)
VALUES ('American');

/* populate participant */
INSERT INTO participant (lname, fname, dob, race_asian, race_native, race_pacific, race_black,
race_white, race_unknown, ethnicity_id)
VALUES ('Surresh',
      'Mohinder',
      '1975-10-10',
      null,
      null,
      null,
      null,
      null,
      null,
      (SELECT ethnicity_id FROM ethnicity WHERE ethnicity_name = 'Hindu'));

INSERT INTO participant (lname, fname, dob, race_asian, race_native, race_pacific, race_black,
race_white, race_unknown, ethnicity_id)
VALUES ('Parkman',
      'Matt',
      '1980-12-01',
      null,
      null,
      null,
      null,
      null,
      'Y',
      null,
      (SELECT ethnicity_id FROM ethnicity WHERE ethnicity_name = 'White'));

INSERT INTO participant (lname, fname, dob, race_asian, race_native, race_pacific, race_black,
race_white, race_unknown, ethnicity_id)

```

```

VALUES ('Nakamura',
        'Hiro',
        '1981-05-18',
        'Y',
        null,
        null,
        null,
        null,
        null,
        (SELECT ethnicity_id FROM ethnicity WHERE ethnicity_name = 'Orient'));

INSERT INTO participant (lname, fname, dob, race_asian, race_native, race_pacific, race_black,
race_white, race_unknown, ethnicity_id)
VALUES ('Sanders',
        'Micah',
        '1989-04-20',
        null,
        null,
        null,
        'Y',
        'Y',
        null,
        (SELECT ethnicity_id FROM ethnicity WHERE ethnicity_name = 'American'));

/* populate event_status */
INSERT INTO event_status (event_status_name)
VALUES ('Scheduled');

INSERT INTO event_status (event_status_name)
VALUES ('Cancelled');

INSERT INTO event_status (event_status_name)
VALUES ('Completed');

INSERT INTO event_status (event_status_name)
VALUES ('In progress');

INSERT INTO event_status (event_status_name)
VALUES ('Anticipated');

/* Populate timepoint_status*/
INSERT INTO timepoint_status (timepoint_status_name)
VALUES ('Scheduled');

INSERT INTO timepoint_status (timepoint_status_name)
VALUES ('Cancelled');

INSERT INTO timepoint_status (timepoint_status_name)
VALUES ('Completed');

INSERT INTO timepoint_status (timepoint_status_name)
VALUES ('In progress');

INSERT INTO timepoint_status (timepoint_status_name)
VALUES ('Undefined');

INSERT INTO timepoint_status (timepoint_status_name)
VALUES ('Anticipated');

/* Populate action_status */
INSERT INTO action_status (action_status_name)
VALUES ('Scheduled');

INSERT INTO action_status (action_status_name)
VALUES ('Cancelled');

INSERT INTO action_status (action_status_name)
VALUES ('Completed');

INSERT INTO action_status (action_status_name)

```

```
VALUES ('In progress');  
  
INSERT INTO action_status (action_status_name)  
VALUES ('Undefined');  
  
INSERT INTO action_status (action_status_name)  
VALUES ('Anticipated');
```

Bibliography

InterBase/Firebird Documentary Team. Firebird ...going where you're going. 2009. 10 May 2009 <<http://www.firebirdsql.org/index.php?op=history>>.

Microsoft Corporation. Model-View-Controller. 2009. 12 May 2009 <<http://msdn.microsoft.com/en-us/library/ms978748.aspx>>.

Orwoll, Eric. Message from the director. 2007. 9 May 2009 <<http://www.octri.org/octri/public/director.aspx?pageid=1>>.

Zarko Gajic. Delphi History from Pascal to Borland Developer Studio 2006. 2009. 10 May 2009 <<http://delphi.about.com/cs/azindex/a/dhistory.htm>>.

Zend Technologies Ltd. About Zend Framework. 2009. 11 May 2009 <<http://framework.zend.com/about/overview>>.