|  |  |
| --- | --- |
|  | Database Design Document |
| 5/7/2013 | Graduate Capstone |

Table of Contents

[1. Introduction 3](#_Toc355695636)

[1.1 Intended Audience 3](#_Toc355695637)

[1.2 References 3](#_Toc355695638)

[1.3 Revision History 3](#_Toc355695639)

[2 Database-Wide Design Decisions 4](#_Toc355695640)

[2.1 Appearance / Naming Convention 4](#_Toc355695641)

[2.1.1 Table Naming Convention 4](#_Toc355695642)

[2.1.2 Column Naming Convention 4](#_Toc355695643)

[2.2 DBMS Platform 4](#_Toc355695644)

[2.3 Operations 4](#_Toc355695645)

[2.4 Maintenance 4](#_Toc355695646)

[3 Detailed Database Design 5](#_Toc355695647)

[3.1 Entity Relationship Diagram 5](#_Toc355695648)

[3.1.1 Entire System 5](#_Toc355695649)

[3.1.2 Focus of Company and Survey 6](#_Toc355695650)

[3.1.3 Focus on Automation 7](#_Toc355695651)

[3.1.4 Focus on Form 7](#_Toc355695652)

[3.1.5 Focus on User Answers 8](#_Toc355695653)

[3.1.6 Focus on User and Language 8](#_Toc355695654)

[3.2 Table Descriptions 9](#_Toc355695655)

[3.2.1 Company Table 9](#_Toc355695656)

[3.2.2 Survey Table 10](#_Toc355695657)

[3.2.3 Revision Table 11](#_Toc355695658)

[3.2.4 Automation Table 12](#_Toc355695659)

[3.2.5 Commands Table 13](#_Toc355695660)

[3.2.6 Command Element Table 14](#_Toc355695661)

[3.2.7 Command Identifier Table 15](#_Toc355695662)

[3.2.8 Form Table 16](#_Toc355695663)

[3.2.9 Form Elements Table 17](#_Toc355695664)

[3.2.10 Form Element Table 18](#_Toc355695665)

[3.2.11 Form Element Attribute Table 19](#_Toc355695666)

[3.2.12 Form Element Option Table 20](#_Toc355695667)

[3.2.13 Display Text Table 21](#_Toc355695668)

[3.2.14 Display Text Translation Table 22](#_Toc355695669)

[3.2.15 Code Type 23](#_Toc355695670)

[3.2.16 Code 24](#_Toc355695671)

[3.2.17 User Table 25](#_Toc355695672)

[3.2.18 User Data Table 26](#_Toc355695673)

[3.2.19 User Setting Table 27](#_Toc355695674)

[3.2.20 Form Submission Table 28](#_Toc355695675)

[3.2.21 Form Answer Table 28](#_Toc355695676)

[3.2.22 User Submission Table 28](#_Toc355695677)

[Appendix 29](#_Toc355695678)

[Appendix A: Code Types 29](#_Toc355695679)

# 1. Introduction

The purpose of this document is to define the tables, fields and attributes of the database.

## 1.1 Intended Audience

This documented is intended for individuals of a high technical background.

## 1.2 References

* <http://wiki.healthmetricsnetwork.info/wiki-kigali/lib/exe/fetch.php?media=templates:databasedesigndocumenttemplate.dot>
* <http://www.mbrs.doe.gov.bz/dbdocs/tech/Design.pdf>
* <https://apps.bsu.edu/AdminConsole/Documentation/SQL/Design/Database.aspx>

## 1.3 Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Date | Reason For Change | Version |
| Andy Bottom | 03/18/2013 | Started Creating the Document; Created documentation about 75% of the tables; | 0.1 |
| Andy Bottom | 04/27/2013 | Created a more formal document; Implemented most of the remaining tables; | 0.2 |
| Andy Bottom | 05/07/2013 | Finalized the layout and added several attributes | 1.0 |
| Andy Bottom | 08/18/2013 | Updated all tables and diagrams to be current with what is currently modeled in the system. | 2.0 |

# 2 Database-Wide Design Decisions

## 2.1 Appearance / Naming Convention

### 2.1.1 Table Naming Convention

All lowercase and words are separated by underscores (\_). Ex. table\_foobar

### 2.1.2 Column Naming Convention

All lowercase and words are separated by underscores (\_). Ex. foobar\_id

## 2.2 DBMS Platform

The database is hosted on a Carroll University Network Server. The platform of the database is Microsoft SQL Server. To manage the server, Microsoft SQL Server Management Studio is the software used.

## 2.3 Operations

Currently there isn’t any system set up currently for backing up the server. As it becomes time to officially release the product, it will be looked into as possible solutions to back-up the system.

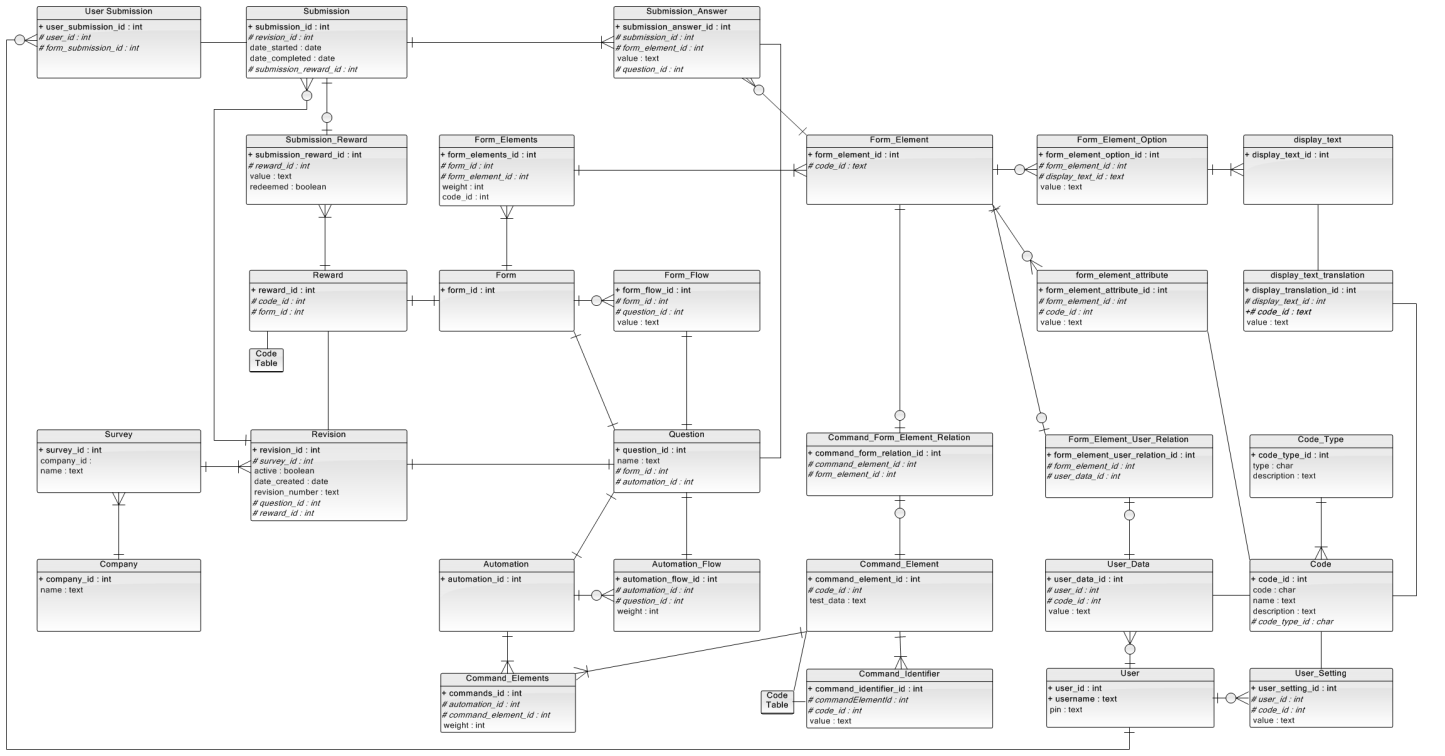
## 2.4 Maintenance

The data in the database will be managed mainly through the Administrator Back-Office Application. Also, monthly, any security updates for the server will be needed to go through and update.

# 3 Detailed Database Design

## 3.1 Entity Relationship Diagram

### 3.1.1 Entire System

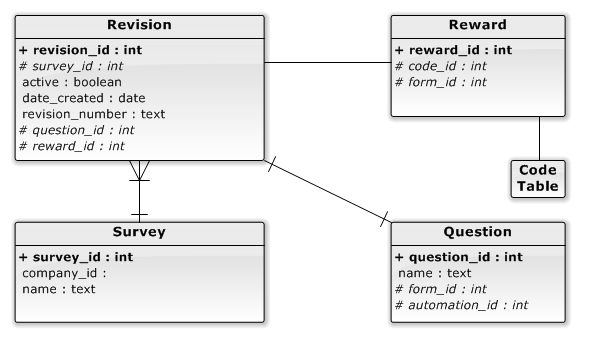


### 3.1.2 Focus of Company and Survey

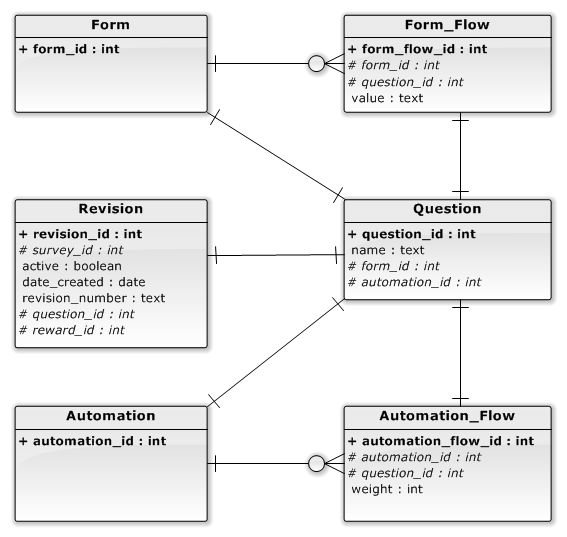
### C:\Users\abottom\Desktop\junk screenshot folder\Entityrelationshipdiagram1.png

@TODO: fix the extra revision id in revision to reward\_id

### 3.1.3 Focus on Revision



### 3.1.4 Focus on Question



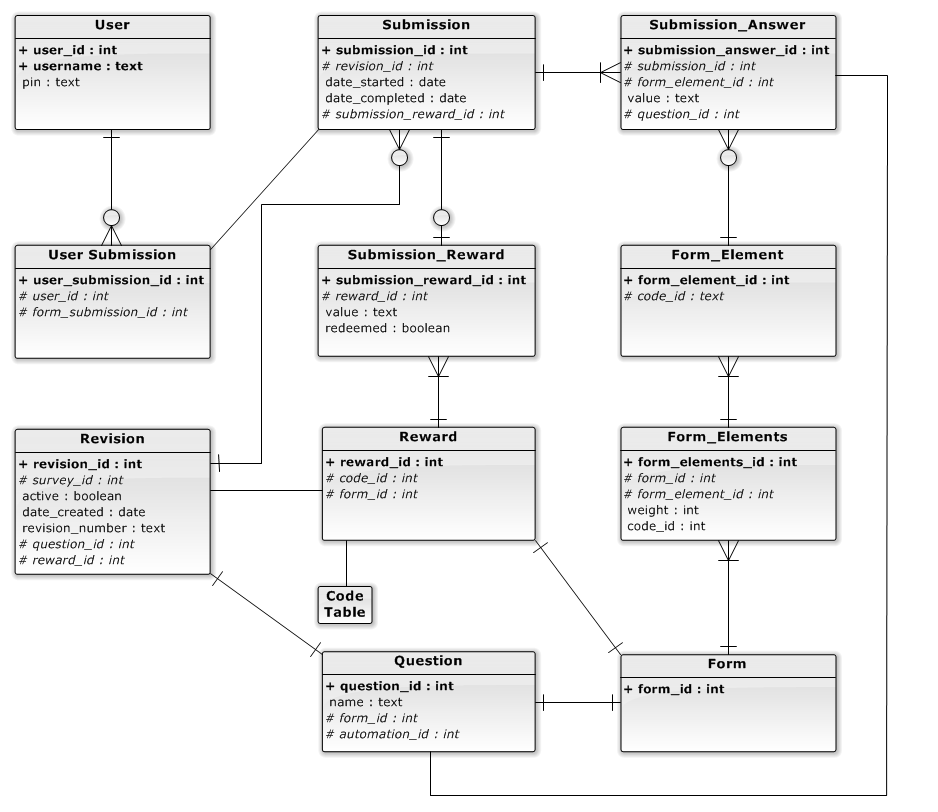
### 3.1.5 Automation

### 

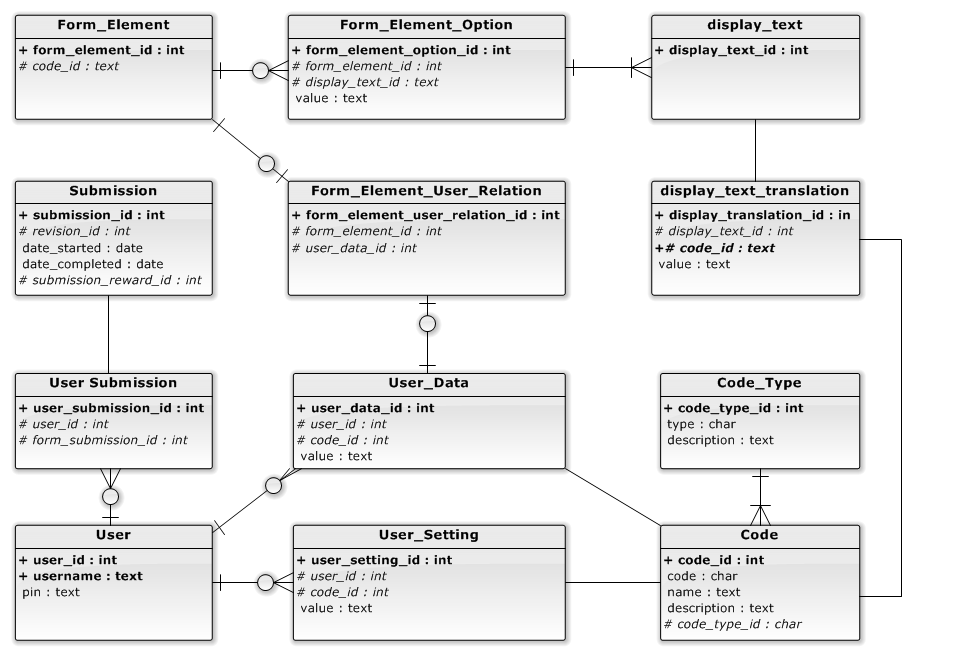
### 3.1.6 Form

### 

### 3.1.7 Focus on Submission and Reward



### 3.1.8 Focus on User and Language



## 3.2 Table Descriptions

### 3.2.1 Company Table

#### 3.2.1.1 Description

The company table is the top table in the theoretical hierarchy in the database. The company table takes the object oriented approach as representing the companies in the project.

There aren't too many attributes in the company table right now; however I will eventually be referencing a company\_attribute table to dynamically add attributes to the table without creating predefined attributes directly into the company table. Further down the implementation, attributes may be transitioned from the attribute\_table to the company table when we find out which attributes will be necessary.

#### 3.2.1.2 Table Info

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column Name | Type | Descriptive Name | Column Properties | Allow Nulls | Description |
| company\_id | int | Company ID | Unique; Index; Primary\_Key; | Not Nullable | The company id is the unique id to identify the individual company |
| name | text | Name |  | Not Nullable | The name of the company |

#### 3.2.1.3 Relationships

* A company can have multiple surveys
* A company may not necessarily always have a survey (although would be a very uncommon occurrence)

### 3.2.2 Survey Table

#### 3.2.2.1 Description

The survey table is the next level underneath companies.

There aren't too many attributes in the survey table right now; however I will eventually be referencing a survey\_attribute table to dynamically add attributes to the table without creating predefined attributes directly into the survey table. Further down the implementation, attributes may be transitioned from the attribute\_table to the survey table when we find out which attributes will be necessary.

#### 3.2.2.2 Table Info

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column Name | Type | Descriptive Name | Column Properties | Allow Nulls | Description |
| survey\_id | int | Survey ID | Unique; Index; Primary\_Key; | Not Nullable | The Survey ID is the unique id to identify the individual survey |
| company\_id | int | Company ID | Index; Foreign\_Key; | Not Nullable | The Company ID that the survey is attached to. |
| name | text | Name |  | Not Nullable | The name of the survey |

#### 3.2.2.3 Relationships

* A survey will always have one company that it is attached to.
* A survey can have multiple revisions
* A survey will always have at least one revision.
* A survey can only have one revision that is active. The others must be set to inactive.

### 3.2.3 Revision Table

#### 3.2.3.1 Description

The revision table contains the objects located underneath surveys. The revision object allows separation from the surveys and the forms and commands. The reason that these are separated from the survey table is to be able to maintain data integrity by keeping the history of the surveys intact, also keeping referential integrity. It is anticipated that survey questions may change. Thus revisions separate the references by adding a middle table away from the survey. ,The revision table references a question id which is the question that starts the flows of both the form and the automation. The reward is a reference to the reward table of what kind of reward is that table. It is not in the survey table because rewards may change, thus the separation.

#### 3.2.3.2 Table Info

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column Name | Type | Descriptive Name | Column Properties | Allow Nulls | Description |
| revision\_id | int | Revision ID | Unique Index; Primary\_Key; | Not Null | The Revision ID is used to identify an individual revision |
| survey\_id | int | Survey ID | Index; Foreign\_Key; | Not Null | Reference to the Survey ID that the revision is attached. |
| active | boolean | Active |  | Not Null | A boolean to determine whether the current revision is the current one. |
| date\_created | DateTime | Date Created |  | Not Null | Stores the date when the revision was created |
| date\_update | DateTime | Date Update |  | Not Null | Stores when the revision was last modified. |
| revision\_number | text | Revision Number |  | Not Null | The human identifiable label to describe the revision. |
| question\_id | int | Question ID | Foreign\_Key; | Not Null | The reference to the question\_id. |
| reward\_id | int | Reward ID | Foreign\_Key; | Not Null | The reference to the reward\_id. |

#### 3.2.3.3 Relationships

* A revision is attached to only one survey.
* A revision has only one reference to a question entry
* A revision has only one reward table reference.

### 3.2.4 Automation Table

#### 3.2.4.1 Description

The automation id is used to provide separation from the revision and the command elements.

#### 3.2.4.2 Table Info

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column Name | Type | Descriptive Name | Column Properties | Allow Nulls | Description |
| automation\_id | int | Automation ID | Unique; Index; Primary\_Key; | Not Null | The Automation ID is the unique id to identify the individual survey |

#### 3.2.4.3 Relationships

* Automation is part of only one question.
* Automation can have none to many automation\_flows associated to it.
* Automation can have none or many command\_elements references associated to it.

### 3.2.5 Command Elements Table

#### 3.2.5.1 Description

The command\_elements table is the relational table that matches a command\_element to an automation\_id.

#### 3.2.5.2 Table Info

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column Name | Type | Descriptive Name | Column Properties | Allow Nulls | Description |
| commands\_id | int | Commands ID | Unique; Index; Primary\_Key; | Not Null | The Commands ID is used to identify an individual reference |
| automation\_id | int | Automation ID | Index; Foreign\_Key; | Not Null | Reference to the Automation ID |
| command\_element\_id | int | Command Element ID | Foreign\_Key; | Not Null | The reference to the command\_element |
| weight | int | Weight |  | Not Null | Weight is the specific priority order of the element in the list |

#### 3.2.5.3 Relationships

* A command\_elements entry will always have only one automation\_id relation
* A command\_elements entry will only have one command\_element relation

### 3.2.6 Command Element Table

#### 3.2.6.1 Description

The command element table represents each individual commands that is listed in the automation aspect of a revision. The command\_element represents all the different type of command\_element, but is identified by the code\_id.

#### 3.2.6.2 Table Info

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column Name | Type | Descriptive Name | Column Properties | Allow Nulls | Description |
| command\_element\_id | int | Command Element ID | Unique; Index; Primary\_Key; | Not Null | The Commands Element ID is used to identify an individual command element |
| code\_id | int | Code ID | Foreign\_Key; | Not Null | The code id is the identifying command\_element\_type |
| test\_data | int | Test Data |  | Null | The input used in tests |

#### 3.2.6.3 Relationships

* The code\_id references a code in the codes table and is the identifier for the type that the command element is.
* A command\_element must have a least one association to an automation table
* A command\_element may have more than one association to an automation table
* A command\_element must have at least one reference to a command\_identifier
* A command\_element may have more than one reference to a command\_identifier

### 3.2.7 Command Identifier Table

#### 3.2.7.1 Description

The command identifier table is used to store the identification type of a command element. This will include the ID Value, Name Value or CSS Value.

#### 3.2.7.2 Table Info

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column Name | Type | Descriptive Name | Column Properties | Allow Nulls | Description |
| command\_identifier\_id | int | Command Identifier ID | Unique; Index; Primary\_Key; | Not Null | Used to identify an individual command identifier |
| command\_element\_id | int | Command Element ID | Index; Foreign Key; | Not Null | Used to associate this identifier to a command element |
| code\_id | int | Code ID | Foreign Key; | Not Null | Used to identify the COMMAND IDENTIFIER TYPE |
| value | int | Value |  | Null | Contains the actual value that will be the identifier. |

#### 3.2.7.3 Relationships

* A code\_identifier will have one and only one code\_id associated. The Code Id will be of COMMAND\_IDENTIFIER\_TYPE value.
* A command\_identifier will have one and only one command\_element that it is associated to.

### 3.2.8 Form Table

#### 3.2.8.1 Description

The form is used to provide separation from the revision and the form elements.

#### 3.2.8.2 Table Info

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column Name | Type | Descriptive Name | Column Properties | Allow Nulls | Description |
| form\_id | int | Form ID | Unique; Index; Primary\_Key; | Not Null | The Form ID is the unique id to identify the individual form |

#### 3.2.8.3 Relationships

* A form entry will always have only one question\_id relation
* A form entry may have none or many form\_elements relation associated to it.
* A form entry may have none or more form\_flow entries associated to it.

### 3.2.9 Form Elements Table

#### 3.2.9.1 Description

The form\_elements table is the relational table that matches a form\_element to a form\_id

#### 3.2.9.2 Table Info

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column Name | Type | Descriptive Name | Column Properties | Allow Nulls | Description |
| form\_elements\_id | int | Form Element ID | Unique; Index; Primary\_Key; | Not Null | The Form Element ID is used to identify an individual reference |
| form\_id | int | Automation ID | Index; Foreign\_Key; | Not Null | Reference to the Form ID |
| form\_element\_id | int | Form Element ID | Foreign\_Key; | Not Null | The reference to the form\_element |
| weight | int | Weight |  | Not Null | Weight is the specific priority order of the element in the list |

#### 3.2.9.3 Relationships

* A form\_element entry will only have one form\_id relation
* A form\_element entry will only have one form\_element relation

### 3.2.10 Form Element Table

#### 3.2.10.1 Description

The form element table represents each individual commands that is listed in the automation aspect of a revision. The form\_element represents all the different type of form\_element, but is identified by the code\_id

#### 3.2.10.2 Table Info

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column Name | Type | Descriptive Name | Column Properties | Allow Nulls | Description |
| form\_element\_id | int | Form Element ID | Unique; Index; Primary\_Key; | Not Null | The Form Element ID is used to identify an individual form element |
| code\_id | int | Code ID | Foreign\_Key; | Not Null | The code id is the identifying form\_element\_type |

#### 3.2.10.3 Relationships

* The code\_id references a code in the codes table and is the identifier for the type that the command element is.
* A form\_element must have a least one association to an form table
* A form\_element may have more than one association to an form table
* A form\_element may have at least one reference to a form\_element\_attribute
* A form\_element may have more than one reference to a form\_element\_attribute
* A form\_element may have at least one reference to a form\_element\_option
* A form\_element may have more than one reference to a form\_element\_option

### 3.2.11 Form Element Attribute Table

#### 3.2.11.1 Description

The form element attribute table contains the attribute fields that will be used for display purposes of that element. For instance, if the form element was a TEXTBOX, then the attribute may be the SIZE property of how long the textbox will display.

#### 3.2.11.2 Table Info

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column Name | Type | Descriptive Name | Column Properties | Allow Nulls | Description |
| form\_element\_attribute\_id | int | Form Element Attribute ID | Unique; Index; Primary\_Key; | Not Null | Used to identify an individual reference |
| form\_element\_id | int | Form Element ID | Index; Foreign\_Key; | Not Null | Reference to the Form Element ID |
| code\_id | int | Code ID | Foreign\_Key; | Not Null | Used to identify the FORM ATTRIBUTE TYPE |
| value | text | Value |  | Not Null | Contains the value of the attribute |

#### 3.2.11.3 Relationships

* A form\_element\_attribute will have one and only one code\_id associated to it. The Code will be of type FORM\_ELEMENT\_ATTRIBUTE\_TYPE.
* A form\_element will have one and only one form\_element associated to it.

### 3.2.12 Form Element Option Table

#### 3.2.12.1 Description

The Form Element Option Table will contain the options that are offered to the user to provide very easy of use. The options will be text values for a TEXTBOX and TEXTAREA field. The will be options for a RADIO, CHECKBOX, SELECTBOX elements.

#### 3.2.12.2 Table Info

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column Name | Type | Descriptive Name | Column Properites | Allow Nulls | Description |
| form\_element\_option\_id | int | Form Element Option ID | Unique; Index; Primary\_Key; | Not Null | Used to identify an individual reference |
| form\_element\_id | int | Form Element ID | Index; Foreign\_Key; | Not Null | Reference to the Form Element ID |
| display\_text\_id | int | Display Text ID | Foreign\_Key; | Not Null | Reference the instance of the Display Text |
| value | text | Value |  | Not Null | Contains the value of the attribute |

#### 3.2.12.3 Relationships

* The form\_element\_option will have one and only one display\_text associated to it.
* The form\_element\_option will have one and only one form\_element associated to it.

### 3.2.13 Display Text Table

#### 3.2.13.1 Description

The display\_text is a table that will hold the single representation of a text for displays.

#### 3.2.13.2 Table Info

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column Name | Type | Descriptive Name | Column Properties | Allow Nulls | Description |
| display\_text\_id | int | Display Text ID | Unique; Index; Primary\_Key; | Not Null | Used to identify an individual reference |

#### 3.2.13.3 Relationships

* display\_text will have one or more display\_text\_translations.
* display\_text can have one or more types of references where it is used.

### 3.2.14 Display Text Translation Table

#### 3.2.14.1 Description

The Display Text Translation is simply a realization of a display\_text for a specific language.

#### 3.2.14.2 Table Info

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column Name | Type | Descriptive Name | Column Properties | Allow Nulls | Description |
| display\_text\_translation\_id | int | Display Text Translation ID | Unique; Index; Primary\_Key; | Not Null | Used to identify an individual reference |
| display\_text\_id | int | Display Text ID | Index; Foreign\_Key; | Not Null | Used to identify the related Display Text Object |
| code\_id | int | Code ID | Foreign\_Key; | Not Null | Identifies the LANGUAGE TYPE |
| value | text | Value |  | Not Null | The value of the Display Text |

#### 3.2.14.3 Relationships

* display\_text\_translation will have one and only one display\_text associated to it.
* display\_text\_translation will have one and only one code associated to it, used to identify the language. The code will be of type LANGUAGE\_TYPE

### 3.2.15 Code Type

#### 3.2.15.1 Description

Code type is a type identifier for constants. It is what groups certain constants into a similar category.

#### 3.2.15.2 Table Info

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column Name | Type | Descriptive Name | Column Properties | Allow Nulls | Description |
| code\_type\_id | int | Code Type ID | Unique; Index; Primary\_Key; | Not Null | Used to identify an individual reference |
| type | text | Type |  | Not Null | Identifies the code\_type but computer\_ identifiable. |
| description | text | Description |  | Not Null | Identifiers the Code Type via human readable |

#### 3.2.15.3 Relationships

* code\_type can have none or more codes associated to it.

### 3.2.16 Code

#### 3.2.16.1 Description

Code is simply a constant that is used in the system.

#### 3.2.16.2 Table Info

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column Name | Type | Descriptive Name | Column Properties | Allow Nulls | Description |
| code\_id | int | Code ID | Unique; Index; Primary\_Key; | Not Null | Used to identify an individual reference |
| code\_type\_id | int | Code Type ID | Index; Foreign\_Key; | Not Null | Used to identify the related code\_type |
| code\_value | text | Code Value | Unique; Index; | Not Null | A computer readable code value |
| name | text | Name |  | Not Null | Name of the code |
| description | text | Description |  | Not Null | Description of the code |

#### 3.2.16.3 Relationship

* code will have one and only one code\_type associated to it
* code can be used by none or more areas of the system.

### 3.2.17 User Table

#### 3.2.17.1 Description

This is the main table that holds the user information. Data in this table are User Objects. Currently the attributes of the user are the credentials of that user. However, in the future, there may be a user\_attribute table in which additional attributes will be stored.

Note that this table currently not implemented as it is outside the scope of this project.

#### 3.2.17.2 Table Info

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column Name | Type | Descriptive Name | Column Properties | Allow Nulls | Description |
| user\_id | int | User ID | Unique; Index; Primary\_Key; | Not Null | Used to identify an individual reference |
| username | text | Username | Unique; Index; | Not Null | The Username of the user |
| pin | text | Pin |  | Not Null | The password for the user |

### 3.2.18 User Data Table

#### 3.2.18.1 Description

User data table is the table that holds the Key information about a user. A key information example would be storing the persons GENDER, (male or female.)

Note that this table currently not implemented as it is outside the scope of this project.

#### 3.2.18.2 Table Info

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column Name | Type | Descriptive Name | Column Properties | Allow Nulls | Description |
| user\_data\_id | Int | User Data ID | Unique; Index;  Primary\_Key; | Not Null | Used to identify an individual reference |
| user\_id | int | User ID | Index; Foreign\_Key | Not Null | Used to identify the related User |
| code\_id | int | Code ID | Index; Foreign\_Key; | Not Null | Represents the USER DATA TYPE |
| value | text | Value |  | Not Null | The value of the data entry |

#### 3.2.18.3 Relationships

* user\_data will have one and only one user\_id associated to it.

### 3.2.19 User Setting Table

#### 3.2.19.1 Description

User Setting is a separate table from the user\_data table, whose purpose is to hold setting information. Such as when was the last time the user was logged in, or if the user is a premium user.

Note that this table currently not implemented as it is outside the scope of this project.

#### 3.2.19.2 Table Info

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column Name | Type | Descriptive Name | Column Properties | Allow Nulls | Description |
| user\_setting\_id | Int | User Setting ID | Unique; Index;  Primary\_Key; | Not Null | Used to identify an individual reference |
| user\_id | int | User ID | Index; Foreign\_Key | Not Null | Used to identify the related User |
| code\_id | int | Code ID | Index; Foreign\_Key; | Not Null | Represents the USER SETTING TYPE |
| value | text | Value |  | Not Null | The value of the data entry |

#### 3.2.19.3 Relationships

* user\_setting will have one and only one user\_id associated to it.

### 3.2.20 Submission Table

#### 3.2.20.1 Description

The submission table represents the Submission objects that the users submit their answers in. The submission object contains a list of the answers. In addition the actual reward that results from the submission of the submission object will be stored in an reward in the object.

#### 3.2.20.2 Table Info

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column Name | Type | Descriptive Name | Column Properties | Allow Nulls | Description |
| submission\_id | int | Submission ID | Unique; Index;  Primary\_Key; | Not Null | Used to identify an individual reference |
| revision\_id | int | Revision ID | Index; Foreign\_Key; | Not Null | Identifies the related Revision entry |
| date\_started | datetime | Date Started |  | Not Null | Is the time that the user started filling out the submission. This is used in case we need to save the answers as a partially filled out. |
| date\_completed | datetime | Date Completed |  | Nullable | Is the time that the submission was fully completed and submitted |
| submission\_ reward\_id | int | Submission Reward ID | Index; Foreign\_Key; | Nullable | Is the reference to the associated reward. If the submission wasn’t automated, then the reward would be null until automation is ran. |

#### 3.2.20.3 Relationships

* Submission has one or more submission\_answers associated to it.
* A submission may or may not have one user\_submission reference associated to it.
* A submission may have only one submission\_reward associated to it.
* A submission must have one and only one reference to a Revision entry.

### 3.2.21 Submission Answer Table

#### 3.2.21.1 Description

The submission answer table is the representation of the Answer object of the submission object. The answer may have a value, which will be what the user inputted. In addition the Answers contain all the question\_ids and form\_elements that the user went through on the form side.

#### 3.2.21.2 Table Info

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column Name | Type | Descriptive Name | Column Properties | Allow Nulls | Description |
| submission\_ answer\_id | int | Submission Answer ID | Unique; Index;  Primary\_Key; | Not Null | Used to identify an individual reference |
| submission\_id | int | Submission ID | Index; Foreign\_Key; | Not Null | Identifies the submission object the answer is associated to |
| form\_element\_id | int | Form Element ID | Index; Foreign\_Key; | Not Null | Identifies which form\_element the answer is submitting for. |
| value | text | Value |  | Null | The value of the answer that the user submitted |
| question\_id | int | Question ID | Index; Foreign\_Key; | Not Null | The question that the answer was answered in. |

#### 3.2.21.3 Relationships

* An answer is a part of only and only one submission entry
* An answer has one and only one question reference associated to it.
* An answer has one and only one form\_element associated to it.

### 3.2.22 Submission Reward Table

#### 3.2.22.1 Description

Submission Reward is the actual Reward that is resultant of a successful submission. The reward is created only when the web driver is ran and completed a submission. The reward has an id indicating what type of reward it is, and the value of the code or coupon.

During the automation, if the form has receipt reward code, then the web driver will find the reward with that form.

#### 3.2.22.2 Table Info

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column Name | Type | Descriptive Name | Column Properties | Allow Nulls | Description |
| submission\_ reward | int | Submission Reward | Unique; Index;  Primary\_Key; | Not Null | Used to identify an individual reference |
| reward\_id | int | Reward ID | Index; Foreign\_Key; | Not Null | The actual realization of a Reward. The identifier for that reward that it is. |
| value | text | Value |  | Null | The value of the reward |
| redeemed | boolean | Redeemed |  | Not Null | Determines whether or not the reward has been redeemed by the user |

#### 3.2.22.3 Relationships

* A submission\_reward has one and only one submission associated to it.
* A submission\_reward has one and only one reward that it follows after.

### 3.2.23 Question Table

#### 3.2.23.1 Description

@TODO: Outside the scope of the current release; To be implemented at a later time.

#### 3.2.23.2 Table Info

@TODO: Outside the scope of the current release; To be implemented at a later time.

#### 3.2.23.3 Relationships

@TODO: Outside the scope of the current release; To be implemented at a later time.

### 3.2.24 Reward Table

#### 3.2.24.1 Description

@TODO: Outside the scope of the current release; To be implemented at a later time.

#### 3.2.24.2 Table Info

@TODO: Outside the scope of the current release; To be implemented at a later time.

#### 3.2.24.3 Relationships

@TODO: Outside the scope of the current release; To be implemented at a later time.

### 3.2.25 Command Form Element Relation Table

#### 3.2.25.1 Description

@TODO: Outside the scope of the current release; To be implemented at a later time.

#### 3.2.25.2 Table Info

@TODO: Outside the scope of the current release; To be implemented at a later time.

#### 3.2.25.3 Relationships

@TODO: Outside the scope of the current release; To be implemented at a later time.

### 3.2.26 Form Flow Table

#### 3.2.26.1 Description

@TODO: Outside the scope of the current release; To be implemented at a later time.

#### 3.2.26.2 Table Info

@TODO: Outside the scope of the current release; To be implemented at a later time.

#### 3.2.26.3 Relationships

@TODO: Outside the scope of the current release; To be implemented at a later time.

### 3.2.27 Automation Flow Table

#### 3.2.27.1 Description

@TODO: Outside the scope of the current release; To be implemented at a later time.

#### 3.2.27.2 Table Info

@TODO: Outside the scope of the current release; To be implemented at a later time.

#### 3.2.27.3 Relationships

@TODO: Outside the scope of the current release; To be implemented at a later time.

### 3.2.28 Form Element User Relation Table

#### 3.2.28.1 Description

@TODO: Outside the scope of the current release; To be implemented at a later time.

#### 3.2.28.2 Table Info

@TODO: Outside the scope of the current release; To be implemented at a later time.

#### 3.2.28.3 Relationships

@TODO: Outside the scope of the current release; To be implemented at a later time.

### 3.2.29 User Submission Table

#### 3.2.29.1 Description

@TODO: Outside the scope of the current release; To be implemented at a later time.

#### 3.2.29.2 Table Info

@TODO: Outside the scope of the current release; To be implemented at a later time.

#### 3.2.29.3 Relationships

@TODO: Outside the scope of the current release; To be implemented at a later time.

# Appendix

## Appendix A: Code Types

|  |
| --- |
| Code Type |
| FORM\_ELEMENT\_TYPE |
| COMMAND\_ELEMENT\_TYPE |
| CODE\_TYPE |
| LANGUAGE\_TYPE |