DATABASE DESIGN DOCUMENT

8/22/2013

Graduate Capstone

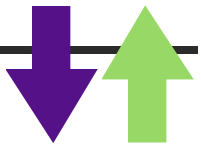
RECEIPT REWARDS

ddedwf

Table of Contents

1. Introduction....4
1.1 Intended Audience....4
1.2 References....4
1.3 Revision History....4
2.1 Appearance / Naming Convention....5
2.1.1 Table Naming Convention....5
2.1.2 Column Naming Convention....5
2.2 DBMS Platform....5
2.3 Operations....5
2.4 Maintenance....5
3.1 Entity Relationship Diagram....6
3.1.1 Entire System....6
3.1.2 Focus of Company and Survey....7
3.1.3 Focus on Revision....7
3.1.4 Focus on Question....8
3.1.5 Automation....9
3.1.6 Form....9
3.1.7 Focus on Submission and Reward....10
3.1.8 Focus on User and Language....11
3.2 Table Descriptions....12
3.2.1 Company Table....12
3.2.2 Survey Table....13
3.2.3 Revision Table....14
3.2.4 Automation Table....15
3.2.5 Command Elements Table....16
3.2.6 Command Element Table....17
3.2.7 Command Identifier Table....18
3.2.8 Form Table....19
3.2.9 Form Elements Table....20
3.2.10 Form Element Table....21
3.2.11 Form Element Attribute Table....22
3.2.12 Form Element Option Table....23
3.2.13 Display Text Table....24
3.2.14 Display Text Translation Table....25
3.2.15 Code Type....26
3.2.16 Code....27
3.2.17 User Table....28
3.2.18 User Data Table....29
3.2.19 User Setting Table....30
3.2.20 Submission Table....31

3.2.21 Submission Answer Table 32
3.2.22 Submission Reward Table 33
3.2.23 Question Table 34
3.2.24 Reward Table 35
3.2.25 Command Form Element Relation Table 36
3.2.26 Form Flow Table 37
3.2.27 Automation Flow Table 38
Appendix 39
Appendix A: Code Types 39



1. Introduction

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

1.1 Intended Audience

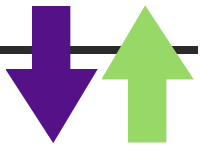
This documented is intended for individuals of a high technical background.

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

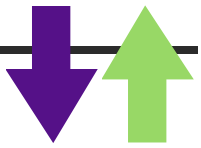
The database is hosted on a Carroll University Network Server. The platform of the database is Microsoft SQL Server. To manage the server, Microsoft SQL Server Management Studio is the software used.

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

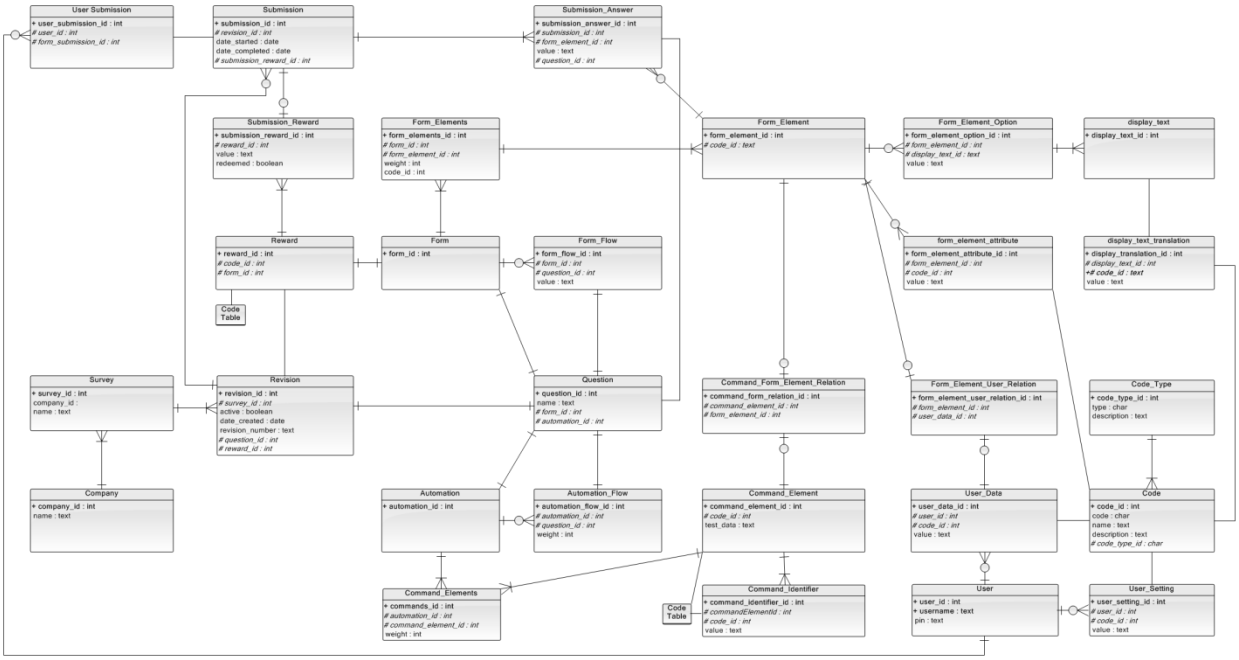
The data in the database will be managed mainly through the Administrator Back-Office Application. Also, monthly, any security updates for the server will be needed to go through and update.



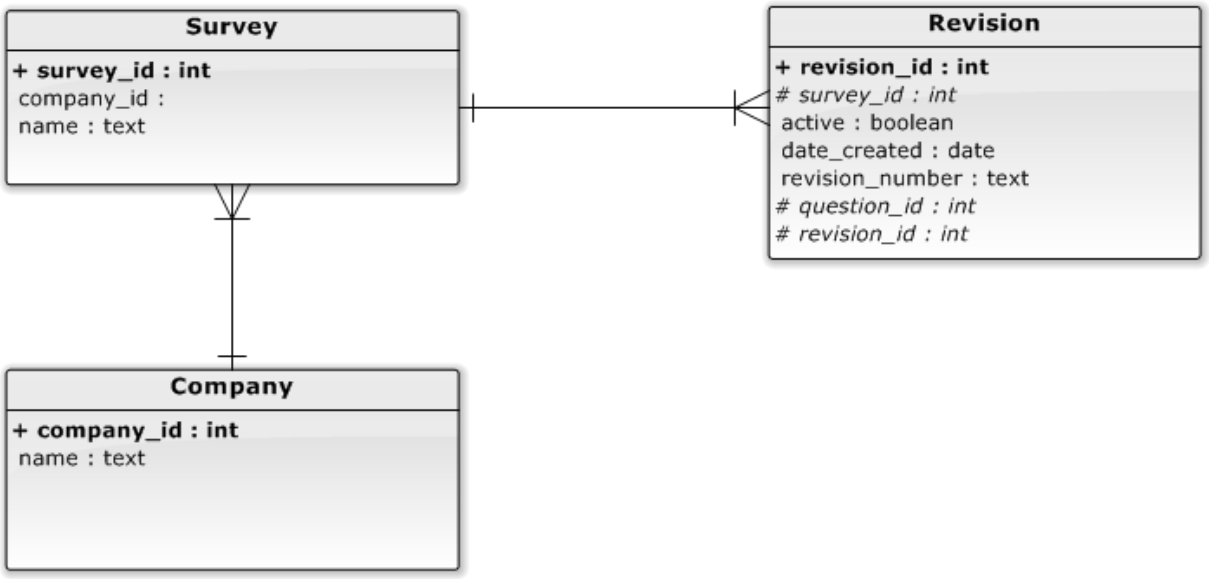
3 Detailed Database Design

3.1 Entity Relationship Diagram

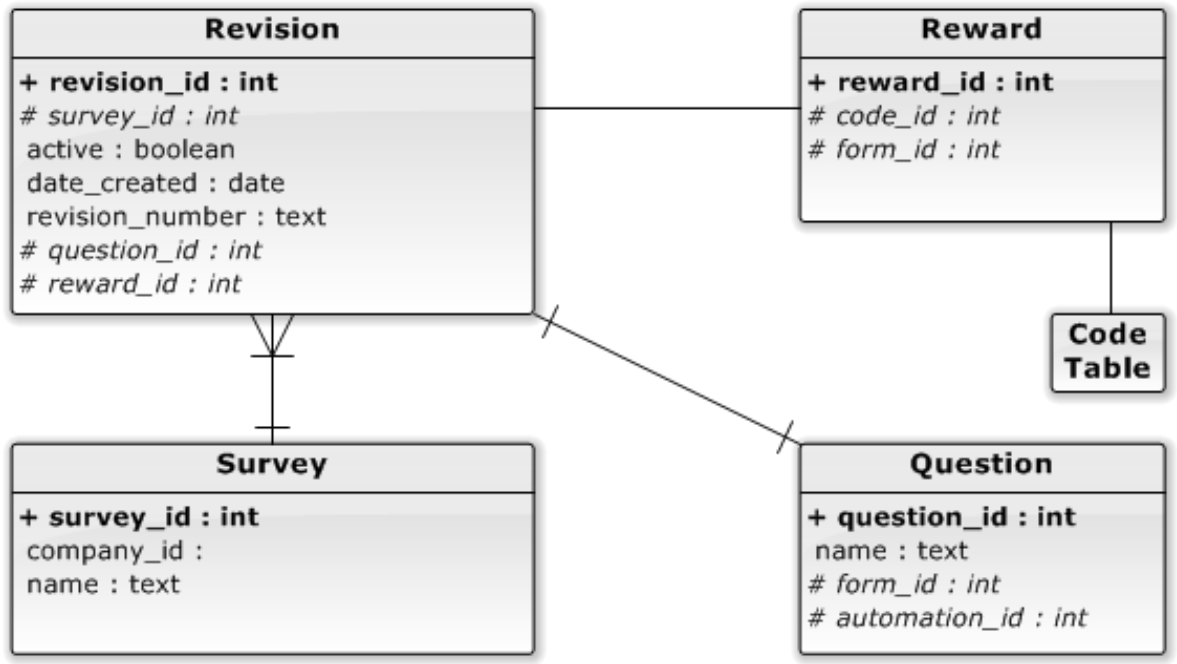
3.1.1 Entire System



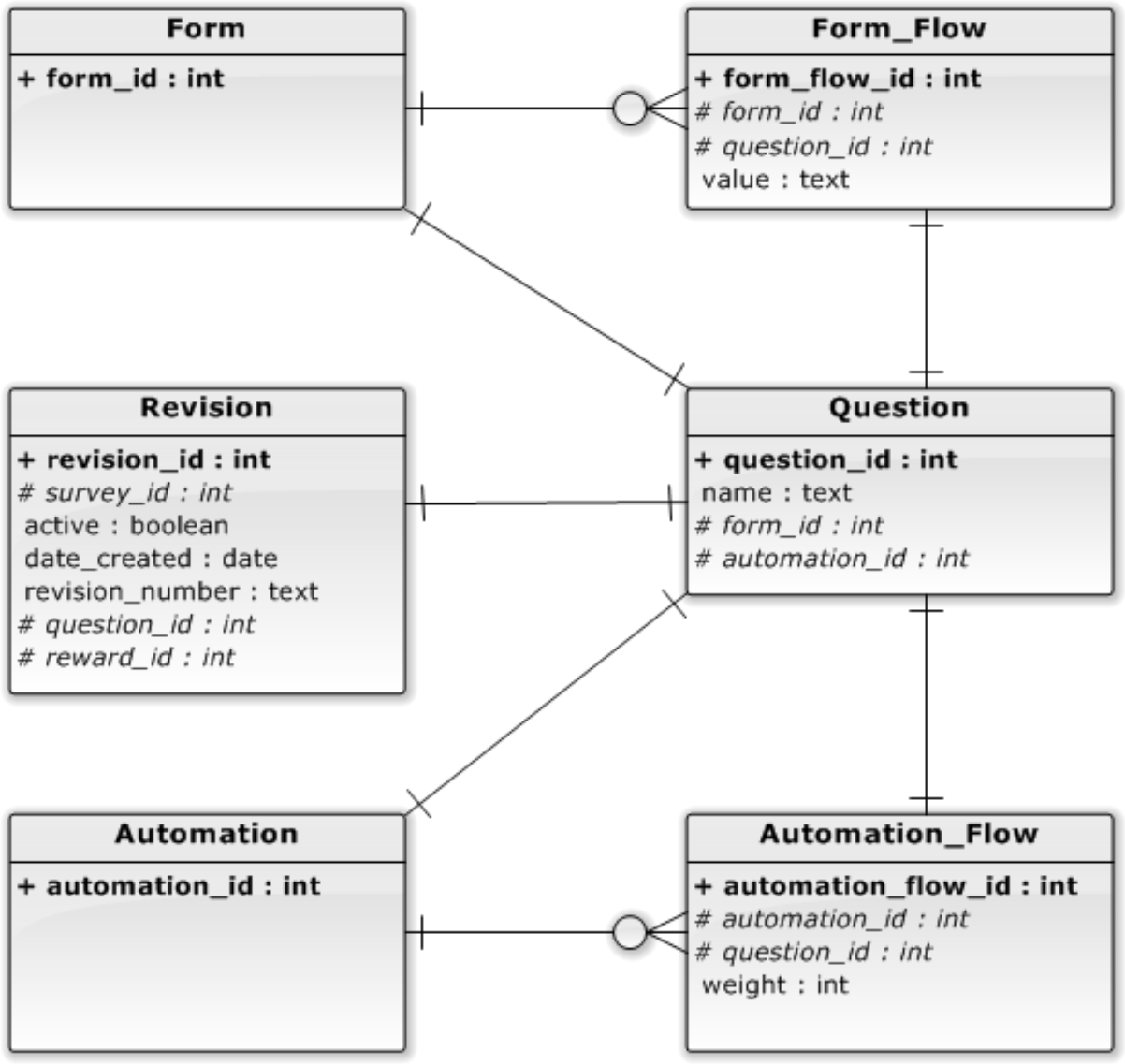
3.1.2 Focus of Company and Survey



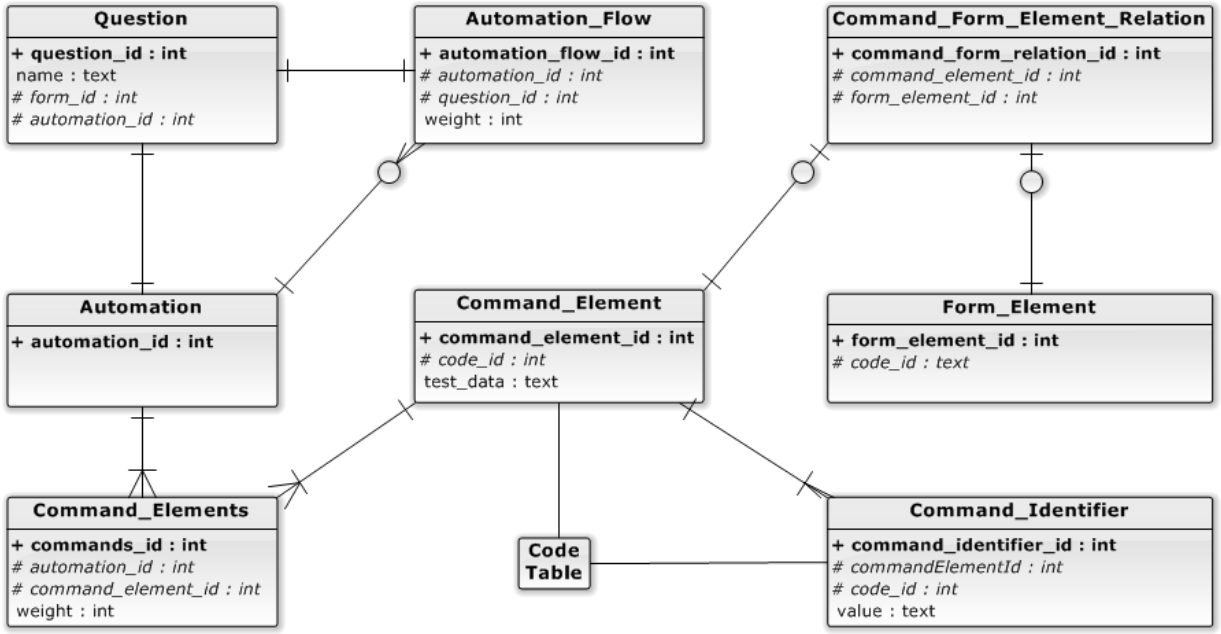
3.1.3 Focus on Revision



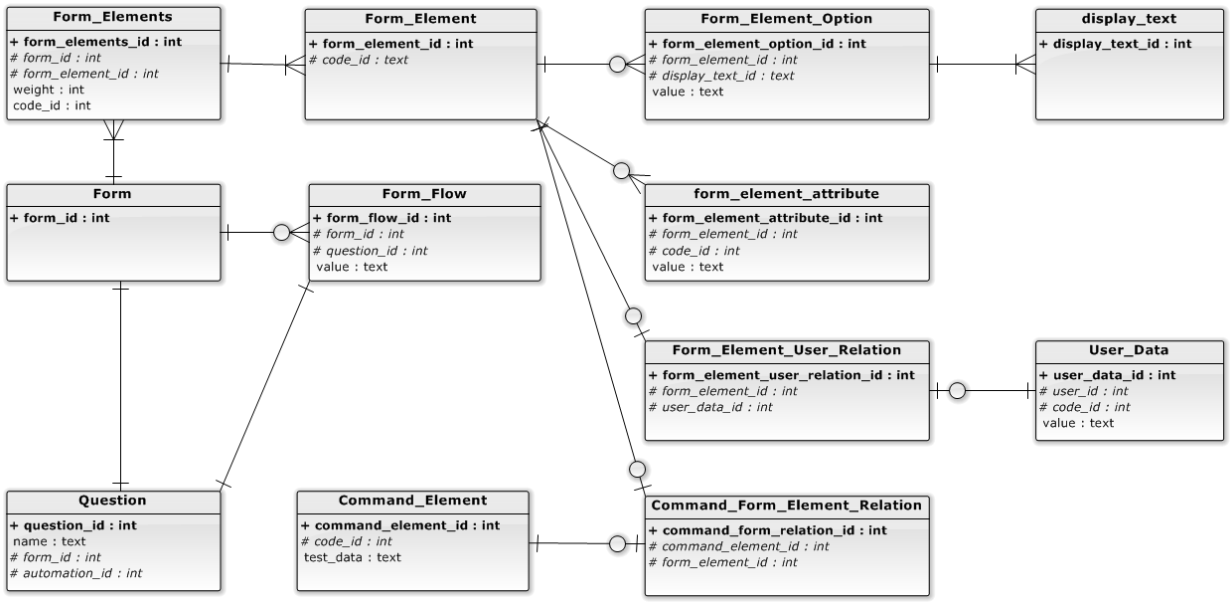
3.1.4 Focus on Question



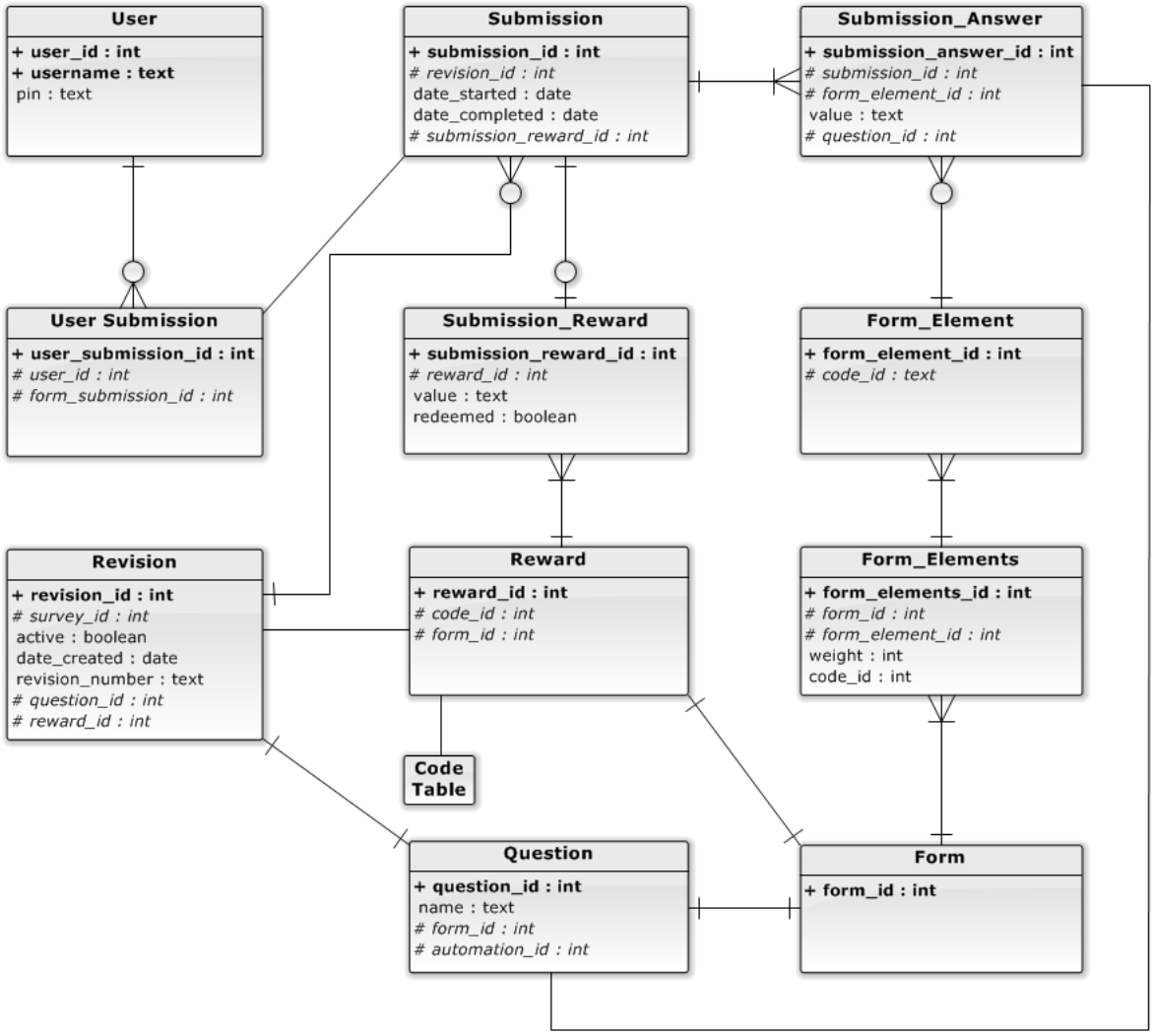
3.1.5 Automation



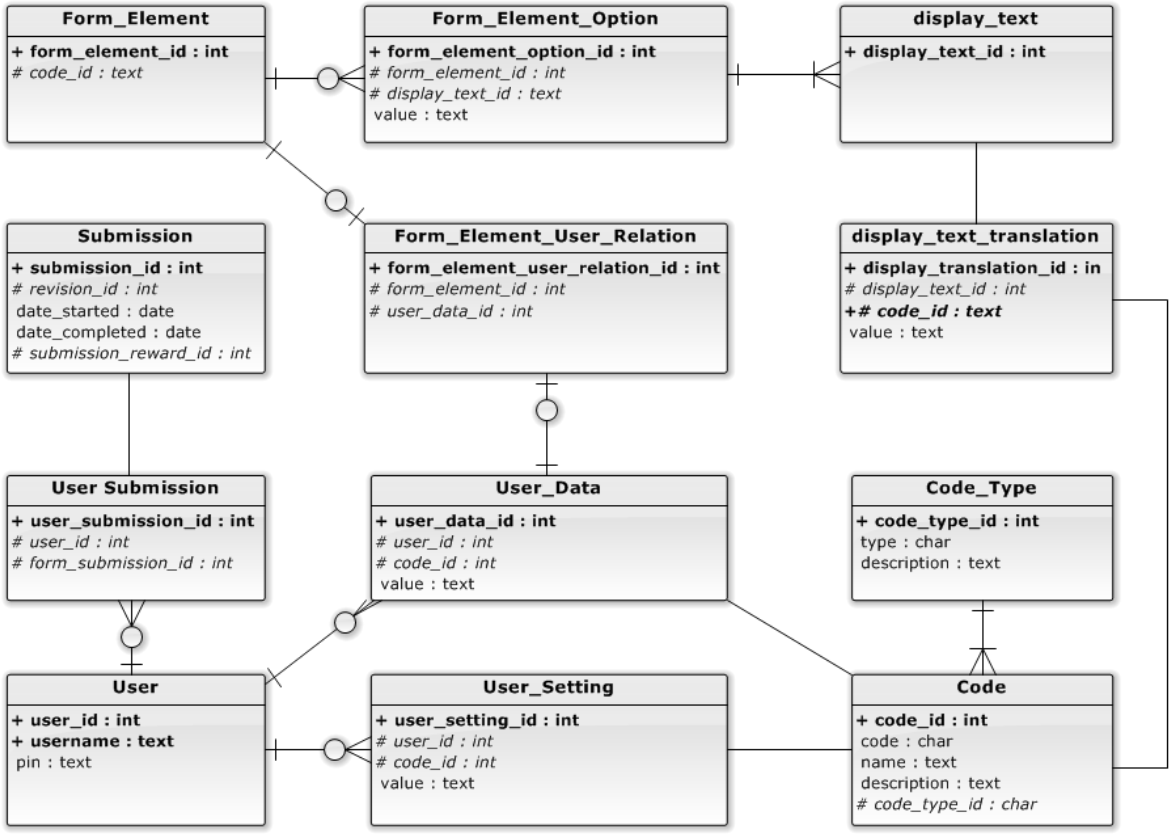
3.1.6 Form



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.

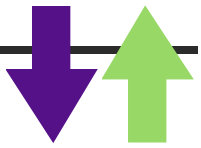
There aren't too many attributes in the company table right now; however I will eventually be referencing a company_attribute table to dynamically add attributes to the table without creating predefined attributes directly into the company table. Further down the implementation, attributes may be transitioned 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.

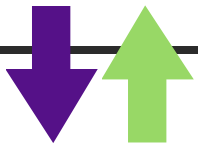
There aren't too many attributes in the survey table right now; however I will eventually be referencing a survey_attribute table to dynamically add attributes to the table without creating predefined attributes directly into the survey table. Further down the implementation, attributes may be transitioned 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.



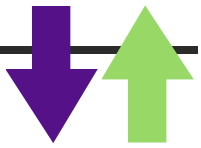
3.2.3 Revision Table

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

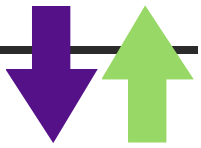
The automation id is used to provide separation from the revision and the command elements.

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

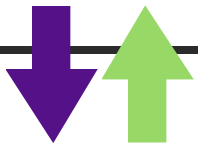
The command_elements table is the relational table that matches a command_element to an automation_id.

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

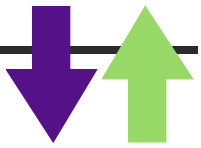
The command element table represents each individual commands that is listed in the automation aspect of a revision. The command_element represents all the different type of command_element, but is identified by the code_id.

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

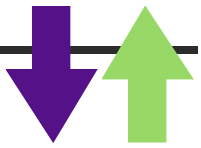
The command identifier table is used to store the identification type of a command element. This will include the ID Value, Name Value or CSS Value. The identifier is how the Web Driver will identify and look for the Web Element on the Satisfaction Survey Website.

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

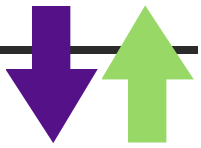
The form is used to provide separation from the question and the form elements.

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

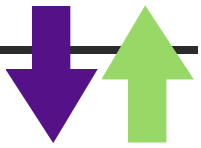
The form_elements table is the relational table that matches a form_element to a form_id

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

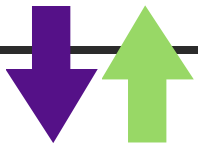
The form element table represents each individual commands that is listed in the automation aspect of a revision. The form_element represents all the different type of form_element, but is identified by the code_id

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

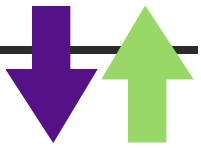
The Form Element Option Table will contain the options that are offered to the user to provide very easy of use. The options will be text values for a TEXTBOX and TEXTAREA field. The will be options for a RADIO, CHECKBOX, SELECTBOX elements.

3.2.12.2 Table Info

Column Name	Type	Descriptive Name	Column Properties	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

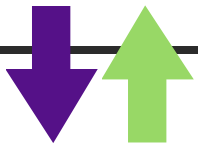
The display_text is a table that will hold the single representation of a text for displays.

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

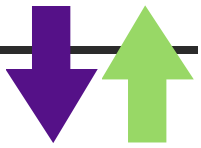
The Display Text Translation is simply a realization of a display_text for a specific language.

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

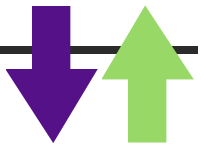
Code type is a type identifier for constants. It is what groups certain constants into a similar category.

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

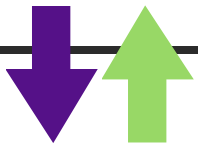
Code is simply a constant that is used in the system.

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

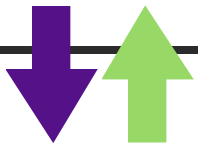
3.2.17.1 Description

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.)

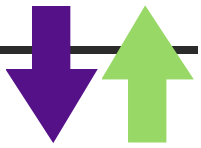
Note that this table currently not implemented as it is outside the scope of this project.

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.

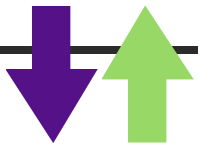
Note that this table currently not implemented as it is outside the scope of this project.

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

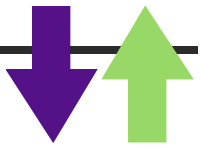
The submission table represents the Submission objects that the users submit their answers in. The submission object contains a list of the answers. In addition the actual reward that results from the submission of the submission object will be stored in an reward in the object.

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		Null	Is the time that the submission was fully completed and submitted
submission_reward_id	int	Submission Reward ID	Index; Foreign_Key;	Null	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

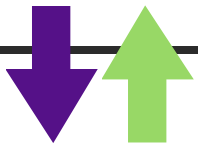
The submission answer table is the representation of the Answer object of the submission object. The answer may have a value, which will be what the user inputted. In addition the Answers contain all the question_ids and form_elements that the user went through on the form side.

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.

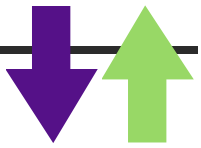
During the automation, if the form has receipt reward code, then the web driver will find the reward with that form.

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

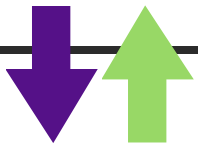
The question table is representation of the Question Objects. Questions are a group, made up of a form and automation. The question has no other properties other than providing the node like structure needed for the multiple flows.

3.2.23.2 Table Info

Column Name	Type	Descriptive Name	Column Properties	Allow Nulls	Description
question_id	int	Question ID	Unique; Index; Primary_Key;	Not Null	Used to identify an individual reference
name	text	Name		Null	This is a meaningful name of which to reference a specific question instance
form_id	int	Form ID	Index; Foreign_Key;	Not Null	The referenced form_id that the question is made up of
automation_id	int	Automation ID	Index; Foreign_Key;	Not Null	The referenced automation_id that the question is composed of

3.2.23.3 Relationships

- One and only one question will be referenced by a single revision entry.
- A question will have one and only one automation entry associated to it.
- A question will have one and only one form entry associated to it.
- A question may have none to many automation_flows which reference it.
- A question may have none to many form_flows which reference it.
- A question may have none to many submission_answers which reference it.



3.2.24 Reward Table

3.2.24.1 Description

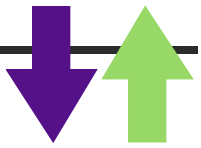
The Reward Table is composed of the Reward objects which are a template for what the reward for a certain revision will be. The submission_rewards will use the Reward entry as a guide to what will be stored and rewarded for that submission.

3.2.24.2 Table Info

Column Name	Type	Descriptive Name	Column Properties	Allow Nulls	Description
reward_id	int	Reward ID	Unique; Index; Primary_Key;	Not Null	Used to identify an individual reference
code_id	int	Code ID	Index; Foreign_Key;	Null	The code id is an indicator as to what type of reward that it is.
form_id	int	Form ID	Index; Foreign_Key;	Not Null	References a form entry

3.2.24.3 Relationships

- A reward will have one and only one revision entry to which it is associated with.
- A reward has one and only one form entry associated to it.
- A reward may be associated to none to many submission_rewards which reference it.
- A reward has one and only one code entry which identifies what type of reward it is.



3.2.25 Command Form Element Relation Table

3.2.25.1 Description

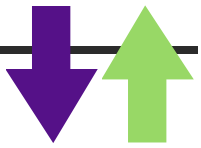
The Command Form Element Relation table is a relation table which only provides the associations between a command_element and a form_element. This association is important because when the web driver is going through the command_elements and a value is needed to be submitted, (such as a textbox,) the web driver will look at all the submission_answers and use the value with the corresponding form_element_id as is stored in the Command Form Element Relation Table.

3.2.25.2 Table Info

Column Name	Type	Descriptive Name	Column Properties	Allow Nulls	Description
command_form_relation_id	int	Command Form Relation ID	Unique; Index; Primary_Key;	Not Null	Used to identify an individual reference
command_element_id	int	Command Element ID	Index; Foreign_Key;	Null	The referenced Command Element
form_element_id	int	Form Element ID	Index; Foreign_Key;	Not Null	The referenced Form Element

3.2.25.3 Relationships

- A command_form_element_relation has one and only one command_element associated.
- A command_form_element_relation has one and only one form_element associated.



3.2.26 Form Flow Table

3.2.26.1 Description

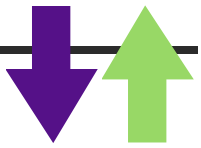
The Form Flow provides the ability for the node (question) to know where the possible next nodes are. Depending on what the user has submitted, it will look for the flow with that value first, otherwise it will use the default flow. This flow is on the form side of the question.

3.2.26.2 Table Info

Column Name	Type	Descriptive Name	Column Properties	Allow Nulls	Description
form_flow_id	int	Form Flow ID	Unique; Index; Primary_Key;	Not Null	Used to identify an individual reference
form_id	int	Form ID	Index; Foreign_Key;	Null	The form that the flow is attached to
question_id	int	Question ID	Index; Foreign_Key;	Not Null	The question that the flow will go next to
value	text	Value		Null	The value that the flow will choose if it matches this value. The default flow will have value as empty

3.2.26.3 Relationships

- A form_flow will have one and only one form associated to it.
- A form_flow will have one and only one question entry to which it points next to.



3.2.27 Automation Flow Table

3.2.27.1 Description

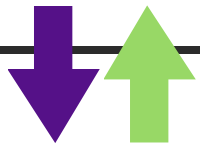
The Automation Flow provides the ability for the node (question) to know where the possible next nodes are. The Automation Flows are prioritized by numerical weight. It will go through the priority and look to see if the question_id is in the array of answers which all have a question_id associated to them. If that question is present, it means that the question was presented to the user and it should then go to that question in the flow. This flow is on the automation side of the question.

3.2.27.2 Table Info

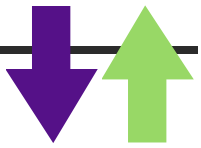
Column Name	Type	Descriptive Name	Column Properties	Allow Nulls	Description
automation_flow_id	int	Automation Flow ID	Unique; Index; Primary_Key;	Not Null	Used to identify an individual reference
automation_id	int	Automation ID	Index; Foreign_Key;	Null	The automation that the flow is attached to
question_id	int	Question ID	Index; Foreign_Key;	Not Null	The question that the flow will go next to
weight	int	Weight		Null	The prioritization of the flows that the web driver will iterate through. Ascending prioritization

3.2.27.3 Relationships

- An automation_flow has one and only one automation entry associated to it.
- An automation_flow has one and only one question entry associated to it.



Appendix

Appendix A: Code Types

Code Type
FORM_ELEMENT_TYPE
COMMAND_ELEMENT_TYPE
CODE_TYPE
LANGUAGE_TYPE