

**National Capital Region Drug Trend Mapping System**

**Detailed Design**

**Prepared For**

**Sgt. Pat Poitevin, RCMP**

**Professor Edmund Strange**

|  |  |
| --- | --- |
| Prepared By: | D. Marcantonio, Project Leader |
|  | L. Desjardins |
|  | E. Dube |
|  | T. Ferch |
|  | L. Morin |
|  | S. Reece |

Submission Date: April 25th, 2011

**List of Figures** Page

1. Context Diagram 4
2. Overview EERD 10
3. Account EERD 11
4. AccountZones EERD 12
5. Authorization EERD 13
6. Drug EERD 14
7. DrugAlias EERD 15
8. DrugColour EERD 15
9. DrugForm EERD 16
10. DrugSideEffect EERD 16
11. DrugType EERD 17
12. Entry EERD 17
13. EntrySideEffect EERD 19
14. Message EERD 20
15. Organization EERD 21
16. PillShape EERD 22
17. PillStamp EERD 22
18. Zone EERD 23
19. AdminPage CD 24
20. Contributors CD 26
21. CreateDataTable CD 27
22. Default CD 27
23. EditEntry CD 28
24. EditOrganization CD 29
25. EditSettings CD 30
26. Entries CD 31
27. Error404 CD 32
28. FindZones CD 32
29. Login CD 33
30. MapService CD 33
31. MasterPage CD 34
32. Messages CD 35
33. NewEntry CD 37
34. Report CD 38
35. ReportHelper CD 40
36. ReportPrint CD 42
37. RequestMebership CD 43
38. ResetPassword CD 44
39. ViewContributors CD 44
40. ViewEntry CD 45
41. ViewOrganization CD 45
42. Welcome CD 46
43. Zones CD 46
44. Admin Page HCL 47
45. Contributors HCI 49
46. Contributors Administrator HCI 50
47. Edit Account HCI 51
48. Edit Entry HCI 52
49. Edit Settings HCI 54
50. Entries HCI 56
51. Entries Administrator HCI 58
52. Login HCI 59
53. Message Center HCI 60
54. New Entry HCI 61
55. Report Preliminary HCI 63
56. Request Membership HCI 65
57. Welcome HCI 66
58. Zones HCI 67

**List of Tables** Page

1. Essential Requirements 5
2. Useful Requirements 7
3. Desirable Requirements 7

**Table of Contents**

Contact Information 1

1. Introduction 2

1.1 Goal and Objectives 2

1.2 Statement of Scope 2

1.2.1 Project Initialization 2

1.2.1.1 Purpose of the Project 2

1.2.1.2 Users of the Product 2

1.2.2 Application and Development Environments 3

1.2.2.1 Application Environment of the Proposed System 3

1.2.2.2 Development Environment of the Proposed System 3

1.2.3 Functional Requirements 4

1.2.3.1 The Scope of the Work-The Proposed Context Diagram 4

1.2.3.2 Functions Provided by the Project 5

1.2.4 Non-Functional Requirements 7

1.2.4.1 Mandated Constraints 7

1.2.4.2 Look and Feel Requirements 7

1.2.4.3 Usability Requirements 7

1.2.4.4 Personalization Requirements 8

1.2.4.5 Ease of Learning Requirements 8

1.2.4.5 Accessibility Requirements 8

1.2.5 Deliverables 8

1.2.5.1 Project Deliverables 8

1.2.5.2 Course Deliverables 8

1.3 Software Context 9

2 Data Design 9

2.1 Session Variables 9

2.2 Database Description 10

2.2.1 Account 11

2.2.2 Account Zones 12

2.2.3 Authorization 13

2.2.4 Drug 14

2.2.5 Drug Alias 15

2.2.6 Drug Colour 15

2.2.7 Drug Form 16

2.2.8 Drug Side Effect 16

2.2.9 Drug Type 17

2.2.10 Entry 17

2.2.11 Entry Side Effect 19

2.2.12 Message 20

2.2.13 Organization 21

2.2.14 Pill Shape 22

2.2.15 Pill Stamp 22

2.2.16 Zone 23

3. Architectural and Component-Level Design 24

3.1 Class Diagrams 24

3.1.1 AdminPage 24

3.1.2 Contributors 26

3.1.3 CreateDataTable 27

3.1.4 Default 27

3.1.5 EditEntry 28

3.1.6 EditOrganization 29

3.1.7 EditSettings 30

3.1.8 Entries 31

3.1.9 Error404 32

3.1.10 FindZones 32

3.1.11 Login 33

3.1.12 MapService 33

3.1.13 MasterPage 34

3.1.14 Message 35

3.1.15 NewEntry 37

3.1.16 Report 38

3.1.17 ReportHelper 40

3.1.18 ReportPrint 42

3.1.19 RequestMembership 43

3.1.20 ResetPassword 44

3.1.21 ViewContributors 44

3.1.22 ViewEntry 45

3.1.23 ViewOrganization 45

3.1.24 Welcome 46

3.1.25 Zones 46

3.2 Software Interface Descriptor 47

3.2.1 Administrator 47

3.2.2 Contributors 49

3.2.3 Contributors (When Logged as Admin) 50

3.2.4 Edit Account 51

3.2.5 Edit Entry 52

3.2.6 Edit Settings 54

3.2.7 Entries 56

3.2.8 Entries (When Logged as Admin) 58

3.2.9 Login 59

3.2.10 Message Centre 60

3.2.11 New Entry 61

3.2.12 Report 63

3.2.13 Request Membership 65

3.2.14 Welcome 66

3.2.15 Zones 67

4. Restrictions Limitations and Constraints 68

Appendix A Data Dictionary

Appendix B Coding Standards

Appendix C Project Gantt chart

Appendix D Detailed Gantt chart for Detailed Design Phase

**Contact Information**

* Client

Sgt./Serg. Pat Poitevin

Coordinator / Coordinateur

Drug and Organized crime Awareness Service

Service de Sensibilisation aux drogues et au crime organisé

RCMP - GRC (Division A)

155 McArthur Avenue

Ottawa, Ontario K1A 0R4

Tel: (613) 993-9380

Fax: (613) 998-7358

e-mail / Couriel: pat.poitevin@rcmp-grc.gc.ca

* Project Group Members

David Marcantonio, Project Leader - davidmarcantonio@gmail.com

Luc Desjardins - luc.r.desjardins@gmail.com

Eric Dube - ericdube88@gmail.com

Tom Ferch - tom.h.ferch@gmail.com

Luc Morin - morinLuc0@gmail.com

Sean Reece - reece.sean@gmail.com

* Project Professor

Edmund Strange

Algonquin College – Woodroffe Campus

Computer Studies Department

1385 Woodroffe Avenue

Ottawa, Ontario

K2G 1V8

(613) 727-4723 ext. 3483

[strange@algonquincollege.com](mailto:strange@algonquincollege.com)

* Project Advisor

Robert Allison

Algonquin College – Woodroffe Campus

Computer Studies Department

1385 Woodroffe Avenue

Ottawa, Ontario

K2G 1V8

(613) 727-4723 ext. 5949

allisor@algonquincollege.com

1. **Introduction**

The detailed design document will provide a detailed overview of the data and interface for the software. Also the documents that will be provided will be the User manual

* 1. **Goal and Objectives**

This section will provide the goals that are to be archived with the software.

* Provided a non scientifically reporting tool to be used to report drug trends in the national capital region.
* Insert a drug entry.
* Contact other users of the system.
* Generate report from zones specified.

* 1. **Statement of Scope**
     1. **Project Initialization**

* + - 1. **Purpose of the Project**

This project aims to address the business needs of Sgt. Pat Poitevin and the Drug and Organized crime Awareness Service of the RCMP (hereto referred to as "the client"). The client wants a system to allow community partners to communicate with him and with each other about trends in drug use in the National Capital Region.

The client has been attempting to assess trends by making telephone calls to the community partners. This does not adequately address the needs of service providers, who need to have accurate information by geographical area. There is an opportunity to provide these community partners with up to date analysis of trends in drug use in the national capital region.

* + - 1. **Users of the Product**

The users of this system will include the client and community partners in the National Capital Region. The community partners who have agreed to participate in submitting information to the system will be able to create an account on the system then submit content and view reports. Community partners may have little computer experience, so the proposed system should be designed as user friendly as possible in order for it to be used with a minimal amount of training.

* + 1. **Application and Development Environments**

This section describes the required hardware and software requirements for both the development and implementation phases of the project.

* + - 1. **Application Environment of the Proposed System**

This section identifies both the hardware and the software that must be in place in the operational (non-testing) environments and configuration details.

1. **Hardware Requirements**

The client will need to provide an ASP.net hosting service for implementation of the completed system. This service can be acquired from many Canadian hosting companies for a reasonable monthly fee which the client has agreed to cover in full. A host can be recommended to the client on request. The service must be acquired at least one week from the implementation date. If the project team does not have access to this service this may introduce a risk of having the implementation date of the project pushed back.

1. **Software Requirements**

The client will not have to provide any software to be used as all software used will be either open source or developed by the project group.

1. **Data Requirements**

The client will supply the project group with the zones used by the Gang Prevention department of the RCMP to be used as the zoning model.

* + - 1. **Development Environment of the Proposed System**

The Waterfall Model Software Engineering paradigm has been chosen for this project because the needs of the client were established.

1. **Hardware Requirements**

The system will need to be developed using an ASP.NET hosting server with storage to host the SQL database. The project group has a hosting server with ASP.NET capabilities and is familiar with its use, it is considered low risk.

1. **Software Requirements**

The system will be developed using the current version of the .NET framework, MSSQL and ASP. The project documentation will be developed using current versions of Microsoft Word, Microsoft Project, Microsoft Visio and Visual Studio. The project group has licensed copies of all this software and is very familiar with their use, they are considered low risk.

1. **Data Requirements**

The client will supply the project group with the zones used by the Gang Prevention department of the RCMP to be used as the zoning model.

1. **Research Requirements**

The project group will have to research the programming style and programs such as ASP.NET with MSSQL databases using the Internet and books as resources.

* + 1. **Functional Requirements**

Functional requirements are functions or features that must be included in the system to satisfy the project needs and be acceptable to the client.

* + - 1. **The Scope of the Work – The Proposed System’s Context Diagram**

The NCRDTMS Context Diagram depicted in Figure 1 contains all of the external entities that produce or consume data that is essential to the operation of the NCRDTMS. As such, the context diagram assists in bounding the scope of the software requirement and also assists in determining the system interfaces.



**Figure 1: Context Diagram**

The NCRDTMS is the web application system that will be developed to meet the requirements of the statement of scope. There are four external entities that can interact with the system. The following descriptions explain the general interaction these entities:

**User:** The user has the privileges to log in to the web application, add, view, and analyze event entries submitted by other users. All accounts will have contact information for users' access.

**Administrator:** The administrator has the same privileges as users, with added features such as editing event entries, and managing user accounts.

**Owner:** The Owner has the ability to manage administrator accounts.

**Pending User:** Pending User has no system access; his account is pending the Owners approval.

* + - 1. **Functions Provided by the Project**

The following section identifies the hardware and software functionality requirements for the project. Each hardware and software requirement listed is prioritized as Essential, Useful, or Desirable as defined below:

* **Essential:** Requirements that must be met by the system to provide a successful project.
* **Useful:** Requirements that would make the system more effective.
* **Desirable:** Requirements that would make the system more attractive to the users.

For each requirement a unique identifier will be used. Requirements will be denoted with the letters “SW”, for software. The following part of identifier will identify a priority classification: “E” for essential, “U” for useful, and “D” for desirable, followed by a numerical value representing the order of priority within the category.

1. **Hardware Functional Requirements**

There are no hardware development activities for this project.

1. **Software Functional Requirements**

The following functional requirements will be addressed by the project.

**Essential Requirements**

|  |  |
| --- | --- |
| SWE1 | The system shall be an internet based client-server application. |
| SWE2 | The system shall allow the user to access the data stored in the system only after having validated their login information. |
| SWE3 | The system shall have 3 types of users - Owner, Administrator, and User. Each type of user has appropriate system access privileges. |
| SWE4 | The system shall associate each User with one or more zones. |
| SWE5 | The system shall allow a User to input Data Entries into any Zone that User is associated with. |
| SWE6 | The system shall allow users to create Generated Reports based on the Event Entries. These Reports shall contain data aggregated by a combination of factors selected by the User including at least:  Date Range  Zone |
| SWE7 | The system shall allow a User to view Event Entries and Generated Reports from any Zone. |
| SWE8 | The system shall provide a graphical representation of each Generated Report through maps and graphs. |
| SWE8 | The system shall include a disclaimer on every Generated Report specifying that the data is anecdotal and not the result of a scientific study. |
| SWE9 | The system shall allow Users to compare Generated Reports across Date Ranges. |
| SWE10 | The system shall allow Administrators all the privileges granted to Users. |
| SWE11 | The system shall allow Administrators to associate or disassociate Users with Zones. |
| SWE12 | The system shall allow Administrators to create new User accounts or delete current Users. |
| SWE13 | The system shall allow Administrators to edit Event Entries in order to correct mistakes. |
| SWE14 | The system shall allow the Owner all privileges granted to Administrators. |
| SWE15 | The system shall allow the Owner to create or delete Administrators. |
| SWE16 | The system shall allow Users to view the Contact Information of every User and send email to any particular User or to all Users. |
| SWE17 | The system shall allow a User to edit their own Contact Information and/or Password. |
| SWE18 | The system shall provide a Periodic Report via email to Users on an opt-in basis. |
| SWE19 | The system shall provide a page showing Users updates since their last login. |

**Useful Requirements**

|  |  |
| --- | --- |
| SWU1 | The system shall provide a message board. |
| SWU2 | The system shall allow Users to save Generated Reports locally. |
| SWU3 | The system shall allow Users to view print-friendly versions of Generated Reports. |

**Table 1: Useful Requirements**

**Desirable Requirements**

|  |  |
| --- | --- |
| SWD1 | The system shall link the map representations of reports to an existing map API. |
| SWD2 | The system shall allow users to save Generated Report Parameters for re-use. |

**Table 2: Desirable Requirements**

* + 1. **Non-Functional Requirements**

Non-functional requirements describe the features, characteristics, and attributes of the system as well as any constraints that may limit the boundaries of the proposed system and the eventual design of the product.

* + - 1. **Mandated Constraints**

This section describes constraints on the requirements and the eventual design of the product.

**Deadline**

The deadline for this project is April 23, 2011.

**Solution Constraints**

The system must be available on the Internet.

The web-interface must be accessible from any web browsing client.

Users must be able to access the web interface through a log in system.

* + - 1. **Look and Feel Requirements**

This section outlines constraints imposed on how the system should be designed for user interaction.

**The Interface**

The system must have a simple web-interface.

* + - 1. **Usability Requirements**

This section details the ability of the User to interact with the system.

**Ease of use**

The web-interface must be simple and intuitive for users that are not technologically inclined.

* + - 1. **Personalization Requirements**

This section describes the way in which the product can be altered or configured to take into account the user’s personal preferences or choice of language.

**Language**

The system will be primarily represented in English, but French may also be implemented, or made so that it can be implemented at a later date. If French is implemented then the user will be able to chose which language they would like displayed. Regardless of implementation the user will always be able to enter information in the language of their choice.

* + - 1. **Ease of Learning Requirements**

The system will be used by users who are technologically inclined and must be presented in an intuitive way as to make users productive within a short period.

* + - 1. **Accessibility Requirements**

The system will follow the Web Content Accessibility Guidelines (WCAG).

* + 1. **Deliverables**

At a minimum, the following constitute the deliverables of the project.

* + - 1. **Project Deliverables**

The following list constitutes the project deliverables:

* Analysis Document set
* Design Document set
* Documented Source Code
* Test Plan and Test Results
* Supporting Manuals - User manuals, Installation manuals
* Tested Client Software in executable format

* + - 1. **Course Deliverables**

The following list constitutes the course deliverables:

* Project Group and Individual Time Logs
* Project Presentation
  1. **Software Context**

The purpose of the software is to aid the client to map drug trends in the capital region. The software will be used strictly for an informative resource to help the client raise awareness of drugs in the region.

1. **Data Design**
   1. **Session Variables**

**-** SessionID (integer)

Each HTML session will be managed by a session variable that is stored

in the MasterPage. The session variable will be instantiated with every successful login.

* 1. **Database Description**



**Figure 2: Overview EERD**

The Data model above is a graphical representation of the database. The database technology that we will be using is MSSQL. The field variables have been modeled around this technology.

* + 1. **Account**



**Figure 3: Account EERD**

**Narrative**:

The Account table will be the resource where the user information will be stored. Account is linked to the Authorization, Message, Organization, AccountZones and Entry.

**Fields**:

int accountID

accountID is the primary key for the account table. The accountID is used in the Message table for the accountID and the receiverAccountID. The accountID is also stored in the AccountZones table for the composite primary key.

The accountID is stored in every entry that is created. accountID cannot be null and will self increment with every new entry in the table.

varchar(50) email

email is the Account email address. email will also be used for the login procedure.

The email will be compared during the login. email cannot be null.

varchar(40) password

The password is used to perform the login authentication procedure. The password will be encrypted using MD5 hashing function before it is inserted into the table. If the email is found the password will be compared to the password stored in the database.

Password cannot be null.

datetime lastLogin

lastLogin is the time that the account was last logged into the system. lastLogin cannot be null.

varchar(50) fullName

The fullName is the name of the creator of the account. fullName will be used for a greeting message at the login.

bit newsLetter

newsLetter is a true or false option if the account wishes to have a newsLetter sent to his email every month. newsLetter will be set during the creating account process and can be changed later in the account settings. newsLetter will be sent to the account email.

bit messageNotify

messageNotify will be used to check if the account wishes to have messages sent directly to his email.

dateTime dateCreated

dateCreated is to keep track of when the account has been created. dateCreated will be submitted by the system at the account creation time. dateCreated cannot be null.

int orgID

The account can belong to an organization. orgID belongs to the Organization table and its primary key. The orgID will be set at the account creation time.

int authID

authID is part of the Authorization table. authID is set to "Pending" at the account creation time. The user will not be able to log into the system. Once the request has been approved the authID will change to "User" The "Owner" can change the "User" Authorization level to "Administrator" to grant that administrator privileges.

* + 1. **AccountZones**



**Figure 4: AccountZones EERD**

**Narrative**:

AccountZones is required as a table in case an Account can be associated to multiple zones. This table enables us to have many accounts associated to many zones.

AccountZones will be required to have composite primary key.

**Fields**:

int accountID

accountID is part of the composite primary key. accountID is associated to the Account table, specifically the accountID field.

int zoneID

zoneID is part of the composite primary key. zoneID is associated to the Zone table, specifically the zoneID field.

* + 1. **Authorization**



**Figure 5: Authorization EERD**

**Narrative:**

The Authorization table will be a pre populated table with the 4 levels of Authorization.

The 4 levels of Authorizations are as followed.

1 Pending, No Access

-Applicant has submitted request to access system.

2 User, Minimal access

-Submit entries

-Review previous entries

-Send messages

-Generate reports

3 Administrator, Medium Access

-The admin can perform all the same task that a user can perform

-Manage Users

-Alter Entries

4 Owner, Full Access

-the owner can perform all the same tasks that an admin can perform

-Change User Privileges

The Authorization table will be pre populated with the 4 authorization levels.

**Fields:**

int authID

authID is the primary key for this table. The primary key will be stored in the Account table. authID cannot be null and will self increment with every entry in the table.

varchar(20) authLevel

authLevel is the word for the level of authorization, ie Pending, User, Administrator and Owner. authLevel cannot be null.

* + 1. **Drug**



**Figure 6: Drug EERD**

**Narrative:**

The Drug table will be the primary source of the drug name. This field is required in the Entry table. Every Drug Alias is associated to one Drug. The Drug table will be pre populated with options for the user for easy selecting. If the option is not listed then the user can add a new entry to the table.

**Field:**

int drugID

The drugID will be the primary key for the Drug table. The drugID will be stored in the Entry table when used. drugID cannot be null and will self increment with every entry.

varchar (50) drugName

The drugName will be associated to the drugID primary key. The drugName will be the drug in plain text. drugName cannot be null.

* + 1. **DrugAlias**



**Figure 7: DrugAlias EERD**

**Narrative:**

The DrugAlias table will be the primary source of a drug’s alternative name. This table will be pre populated with options for the user for easy selecting. If the option is not listed then the user can add a new entry to the table. Every aliasID will be associated to a drugID so that the list of names will be narrowed when selecting.

**Field:**

int aliasID

The aliasID will be the primary key for the Alias table. The aliasID will be stored in the Entry table when used. aliasID cannot be null and will self increment with every entry in the DrugAlias table.

varchar (50) aliasName

The aliasName will be associated to the aliasID primary key. The aliasName will be the keyword or street name of the drug. aliasName cannot be null.

int drugID

The drugID is the linked to the Drug table so that every drug alias belongs to a drug, this makes for easy selecting. drugID cannot be null.

* + 1. **DrugColour**



**Figure 8: DrugColour EERD**

**Narrative:**

The DrugColour table will be the part of the Entry table. DrugColour will describe the color of the drug. This table will be pre populated with options for the user to easily select the right option. The user will also have the opportunity to add colors to the table if the color is not listed.

**Field:**

int colourID

colourID will be the primary key for the table. colourID will be stored in the Entry table. colourID cannot be null and will self increment with every new entry in the table.

varchar(25) colourName

colourName is the text of the color of the drug. colourName will be associated to the primary key in the table.

* + 1. **DrugForm**



**Figure 9: DrugForm EERD**

**Narrative:**

The DrugForm table is the primary resource for the format of drugs. DrugForm is made reference by the Entry table. DrugForm will be pre populated with format of previous entries so that we can auto populate a drop down menu.

**Fields:**

int formID

formID is the primary key for the DrugForm table. formID cannot be null and will self increment with every new entry in the table.

varchar(20) formName

formName will be associated to a formID. formName will be the name of the effect in plain text.

* + 1. **DrugSideEffect**



**Figure 10: DrugSideEffect**

**Narrative:**

The DrugSideEffect table is the primary resource for the side effects of the drugs. DrugSideEffect is made reference by the EntrySideEffect table. DrugSideEffect will be pre populated with side effects of previous entries so that we can auto populate a drop down menu.

**Fields**:

int sideEffectID

sideEffectID is the primary key for the DrugSideEffect table. sideEffectID cannot be null and will self increment with every new entry in the DrugSideEffect table.

varchar(50) sideEffectName

sideEffectName will be associated to a sideEffectID. sideEffectName will be the name of the side effects in plain text.

* + 1. **DrugType**



**Figure 11: DrugType EERD**

**Narrative:**

The DrugType table will be the primary source of the type of drug. This table will be pre populated for the user for easy selecting. If the option is not listed then the user can add a new entry to the table.

**Field:**

int typeID

The typeID will be the primary key for the DrugType table. The typeID will be stored in the Entry table when used. typeID cannot be null and will self increment with every entry.

varchar (20) typeName

The typeName will be associated to the typeID primary key. The typeName will be the keyword of the type of drug. typeName cannot be null.

* + 1. **Entry**



**Figure 12: Entry EERD**

**Narrative:**

The Entry table will be the main source of information that will be reviewed. A user can log into the system and generate drug entries. This table will be the main resource of system. The reports will query this table for all the information to generate the reports.

**Fields:**

int entryID

entryID is the primary key and will be associated to when performing reports. entryID is referred to in the EntryEffect and EntrySideEffect table. entryID cannot be null and will self increment with every new entry in the Entry table.

Varchar(10) age

age will be the age for the entry is based on. Age will have range classifications for easy selecting.

Varchar(8) gender

gender is the gender of the person the entry in based on.

gender can have 3 valid answers,

Male for male

Female for female

Unknown is if the user did not specify the gender of the person

datetime dateUploaded

dateUploaded is the date that the entry was submitted. dateUploaded will be populated by the system. dateUploaded cannot be null.

datetime dateEvent

dateEvent is the approximate date the drug usage event occurred. dateEvent cannot be null.

varchar(1024) comment

comment is if the user would like to add additional comments to the entry.

int accountID

accountID is the account that created the entry. accountID is linked to the Account table and linked to the accountID field. accountID cannot be null.

int zoneEventID

zoneEventID is the location that the current user of the system is situated.

zoneEventID will be pre populated by the system. zoneEventID will link to the Zone table and link to the zoneID field in the table. zoneEventID cannot be null.

int drugAliasID

drugAliasID is the alias of the drug or the street name of the drug.

drugAliasID will link to the DrugAlias table and will link to the aliasID field.

int drugID

drugID is the drug that was used during the drug usage event. drugID is linked to the Drug table and linked to the drugID field. drugID cannot be null.

int typeID

typeID is the type of drug that was used during the event. ie depressant, stimulant and ect. typeID will link to the DrugType Table and will link to the typeID field.

int formID

formID is the format of drug that was used during the event. ie liquid, pill and ect. formID will link to the DrugForm Table and will link to the formID field.

int colourID

colourID is the colour of drug that was used during the event. ie Blue, Yellow and ect. colourID will link to the DrugColour Table and will link to the colourID field.

int stampID

stampID is the stamp of drug that was used during the event. ie star, dolphin and ect. stampID will only be set when the formID is set to pill form. stampID will link to the PillStamp Table and will link to the stampID field.

int shapeID

shapeID is the shape of drug that was used during the event. ie round, square and ect. shapeID will only be set when the formID is set to pill form. shapeID will link to the PillShape Table and will link to the shapeID field.

* + 1. **EntrySideEffect**



**Figure 13: EntrySideEffect EERD**

**Narrative:**

EntrySideEffect is required if an Entry has more than one SideEffect associated to it. EntrySideEffect will contain a composite primary key. EntrySideEffect will link the Entry and DrugSideEffect tables.

**Fields:**

int entryID

entryID is part of the composite primary key. entryID is associated to the Entry table, specifically the entryID field.

int drugSideEffectID

drugSideEffectID is part of the composite primary key. drugSideEffectID is associated to the DrugSideEffect table, specifically the drugSideEffectID field.

* + 1. **Message**



**Figure 14: Message EERD**

**Narrative:**

The message table will be the main resource for the messaging center. The message inserted into the table will be message submitted by the user. The message can either be private messages or public, bulletin board messages.

Note: The bulletin board messages will be sent to an account named bulletin board.

This account is a "dummy" account that all users can read its messages.

Note: There will also be a "Deleted" dummy account that messages that have been sent from a deleted account belong to.

**Fields:**

int messageID:

The primary key for the message this will enable the message center to pull messages from the table. messageID cannot be null and will self increment with every entry in the Message table.

varchar (2048) message

The message sent to the user. The message will be the text being sent to the user.

datetime timeSubmitted

timeSubmitted is the time the message is sent to the user. This field will be populated by the system. timeSubmitted cannot be null.

bit messageRead

The messageRead field will mark the message as being read or not. Once the message is received this field is 1 and once the user has selected the message this will change to 0. The messageRead will change the user interface.

int senderAccountID

senderAccountID is the primary key of the account that is sending the message. The account that is sending the message its primary key will be stored in this field. senderAccountID cannot be null.

int receiverAccountID

receiverAccountID is the primary key of the account that the message is addressed to. The primary key of the account will need to be researched before this field is populated. receiverAccountID cannot be null.

* + 1. **Organization**



**Figure 15: Organization EERD**

**Narrative:**

The organization table will be the table that all the organizations of the system will be stored. The organization could be the company that the account holder belongs to. For example: Shepherds of Good Hope. The account could belong to no organization. This table is populated when a user creates their account. If the organization has not been used before, the user will add a new one.

**Fields:**

int orgID

organizationID is the primary key for the table. The primary key will be stored in the Account table. The primary key is associated to the name of the organization.

organizationID cannot be null and will self increment with every entry in the Organization table.

varchar(50) orgName

organizationName is the name of the organization. The organization name will be stored in this field. organizationName cannot be null.

varchar(256) orgAddress

orgAddress is the address of the organization. The organization address will be stored in this field.

varchar(128) orgWeblink

orgWeblink is the organizations website if they have one. The organization weblink will be stored in this field.

* + 1. **PillShape**



**Figure 16: PillShape EERD**

**Narrative:**

The PillShape table will be the primary source of the Shape of drug. This table will be pre populated for the user for easy selecting. If the option is not listed then the user can add a new entry to the table. PillShape will only be set when an Entry’s formID is set to pill.

**Field:**

int shapeID

The shapeID will be the primary key for the PillShape table. The shapeID will be stored in the Entry table when used. shapeID cannot be null and will self increment with every entry.

varchar (20) shapeName

The shapeName will be associated to the shapeID primary key. The shapeName will be the keyword of the shape of the pill. shapeName cannot be null.

* + 1. **PillStamp**



**Figure 17: PillStamp EERD**

**Narrative:**

The PillStamp table will be the primary source of the stamp of drug. This table will be pre populated for the user for easy selecting. If the option is not listed then the user can add a new entry to the table. PillStamp will only be set when an Entry’s formID is set to pill.

**Field:**

int stampID

The stampID will be the primary key for the PillStamp table. The stampID will be stored in the Entry table when used. stampID cannot be null and will self increment with every entry.

varchar (50) stampName

The stampName will be associated to the stampID primary key. The stampName will be the keyword of the stamp of the pill. stampName cannot be null.

* + 1. **Zone**



**Figure 18: Zone EERD**

**Narrative:**

The Zone table is the primary resource of the zones being used by the DTMS. The Zone table will be pre populated with pre defined zones. The Zone table is referenced by the UserZones table an Account can be associated to more than 1 zone. The Zone table is also referenced by the Entry table, ZoneEventID.

**Fields:**

int zoneID

zoneID is the primary key to the Zone table. zoneID cannot be null and will self increment with every new entry in the Zone table.

varchar(50) zoneName

zoneName is the name associated to the zone. zoneName is required to be associated to a zoneID. zoneName cannot be null.

varchar(1024) zoneBoundaries

zoneBoundaries is the description of zone. zoneBoundaries is required to be associated to a zoneID. zoneBoundaries cannot be null.

1. **Architectural and Component-Level Design**
   1. **Class Diagrams**
      1. **AdminPage**



**Figure 19: AdminPage CD**

The AdminPage Class will handle actions needed by the Administrator to change the entries and the columns in the database. This will include loading the list of current Drugs, Drug Street Name, Drug Form, Drug Type etc.

**Attributes**

Public DTMSDataContext dtmsdc: the linq to sql layer for the database objects.

**Function Algorithms**

**Page\_Load():** This method handles the actions required when the Admin page loads.

**NavigateToEditPage():** This method navigates to the appropriate editing page for detailed editing.

* + 1. **Detailed Edit Page**



**Attributes**

Public DTMSDataContext dtmsdc: the linq to sql layer for the database objects.

**Function Algorithms**

**Page\_Load():** This method handles the actions required when the Admin page loads.

**SearchByName():** This method searches for items based on their name field.

**SearchByURL():** This method searches for items based on their URL.

**SearchByImage():** This method searches for items based on their image name

**SearchByDescription():** This method searches for items based on their description

**EditItem():** Edits the selected item.

**ViewItem():** Views the selected item.

**DeleteItem():** Deletes the selected item.

* + 1. **Contributors**



**Figure 20: Contributors CD**

The ContributorsPage Class will handle actions needed by the ContributorsPage. This will include loading the list of Contributors for display. In addition, it will allow the user to click on any Contributor in the list to send a message to them.

**Attributes**

Public DTMSDataContext dtmsdc: the linq to sql layer for the database objects.

String sorting: Sort direction used for the grid view

**Function Algorithms**

**ContributorGridView\_OnRowCommand():** This method handles the actions sent when a button is pressed.

**ContributorGridView\_PageIndexChanging():** This method handles the actions sent when a button is pressed.

**Filter\_Button\_Click():** This method handles the actions sent by the filter button.

**List\_Changed():** This method handles the actions sent when a list is changed.

**Contributors\_Sorting():** This method handles the actions sent by the headers when sorting.

**Page\_Load():** This method handles the actions when the page loads.

**UpdateView():** This method updates the Gridview with data from the database.

* + 1. **CreateDataTable**



**Figure 21: CreateDataTable CD**

The CreateDataTable Class will handle actions needed to convert Enumerable to DataTables.

**Attributes**

None

**Function Algorithms**

**ToDataTable():** Convert IEnumerable to DataTables.

* + 1. **Default**



**Figure 22 Default CD**

The Default Class will handle actions needed to greet first time potential Contributors to the website. This will give the potential Contributor the option to request membership to the system or to login.

**Attributes**

Public DTMSDataContext dtmsdc: the linq to sql layer for the database objects.

**Function Algorithms**

**Page\_Load():** On page load, if the user is authenticated, redirect to the Welcome page.

* + 1. **EditEntry**



**Figure 23: EditEntry CD**

The EditEntry class will handle actions needed by the Edit Entry Page. Only Administrators and Owners will have access to this page. It will contain the information from a Entry and will have editable fields that the Admin or Owner can change. Once submitted, the information will be stored in the database, replacing the old information.

**Attributes**

Public DTMSDataContext dtmsdc: the linq to sql layer for the database objects.

**Function Algorithms**

**Page\_Load():** Page load will pre fill all lists and pre select the proper one.

**Submit\_Button\_Click():** Runs when the user submits the changed content. This method will update the appropriate Entry.

**DrugChage():** This event handler is called whenever the user selects a different drug from the drop down. It will fill in the drug alias drop down with this drug's associated street names.

**OtherCheck():** This event handler is called when the user changes the selected index of any drop down on this page except for drug. This method will check if the selected value of any drop down is "Other" and will make the text boxes visible/ hidden as needed.

**AddSideEffect\_Button\_Click():** This event handler is called when the user enters a new side effect and hits the add button. The side effect is added to the “added” side of the list.

**RemoveSideEffect\_Button\_Click():** This event handler is called when the user enters a new side effect and hits the remove from list button. The side effect is removed from the “added” side of the list.

**Delete\_Entry():** This method is called when the user hits the "delete" button. This entry and all side effects referenced by it are deleted from the database.

* + 1. **EditOrganization**



**Figure 24: EditOrganization CD**

The EditOrganization Class will handle actions needed to edit the Organization selected. The only contributors that will be able to access this page are administrators and owners.

**Attributes**

Public DTMSDataContext dtmsdc: the linq to sql layer for the database objects.

**Function Algorithms**

**Page\_Load():** Page load fills in all the values for the organization.

**Submit\_Button\_Click():** This method is called when the user hits "submit". It updates the associated Organization to the values entered by the user.

**Delete\_Button\_Click():** This method is called when the user clicks the "delete" button. It will check if the organization has any accounts associated to it and notify the user that it can't be deleted if so, otherwise the organization is deleted.

* + 1. **EditSettings**



**Figure 25: EditSettings CD**

The EditSettings class will handle actions needed to edit the contributors Account Page. This will include displaying the users account information and have the ability to modify this information. This class will allow the user to save the information which will store the new updated information into the database. This page will also give the Admins the ability to change a Users information, such as password and zone information.

**Attributes**

Public DTMSDataContext dtmsdc: the linq to sql layer for the database objects.

**Function Algorithms**

**Page\_Load():** Page load will populate and select all the values for the specified user. If a query string is not specified then a redirect will occur with a query string displaying the user's page.

**Submit\_Button\_Click():** This method will update the account with the updated settings entered by the user.

**Organization\_DropDownList\_SelectedIndexChanged():** This method will occur when the user changes the selected index of the organization drop down. It just checks if "Other" was selected and displays a textbox if it was.

**SendUserEmail():** This method will send an email to the user when their account has been activated by the owner.

* + 1. **Entries**



**Figure 26: Entries CD**

The Entries Class will handle actions required to view all the entries in the database. The page will have filter options and sorting capabilities.

**Attributes**

Public DTMSDataContext dtmsdc: the linq to sql layer for the database objects.

Public String sorting: Sort direction used for the grid view

**Function Algorithms**

**EntriesGridView\_OnRowCommand():** This method handles the actions sent when a button is pressed

**EntriesGridView\_PageIndexChanging():** This method handles the actions sent when a button is pressed

**Entries\_Sorting():** This method handles the actions sent by the headers when sorting

**Filter\_Button\_Click():** This method handles the actions sent by the filter button

**List\_Changed():** This method handles the actions sent when a list is changed

**Page\_Load():** This method handles the actions when the page loads

**UpdateView():** This method updates the Gridview with data from the database.

**Edit():** Navigates the user to the edit page for the specified item.

**View():** Navigates the user to the view page for the specified item.

**Delete():** Deletes the selected item from the list and database.

* + 1. **Error404**



**Figure 27: Error404 CD**

The Error404 Class will handle actions needed to load an error page when there are errors on the page.

**Attributes**

None

**Function Algorithms**

**Page\_Load():**This method handles the actions required when the Error404 page loads.

**Retry\_Link\_Redirect():** This method handles the actions required when the Retry link is clicked.

* + 1. **FindZones**



**Figure 28: FindZones CD**

The FindZones Class will handle actions required to parse all the Zones.txt file to retrieve all the longitude and latitudes from each zone.

**Attributes**

None

**Function Algorithms**

**PopulateZones():** This function parses the Zones.txt file and will find all the coordinates for the Zones

* + 1. **Login**



**Figure 29: Login CD**

The Login Class will handle actions needed by the Login Page. It securely allows a registered user to access the website, and provides a link for non-registered users to request membership to the page.

**Attributes**

Public DTMSDataContext dtmsdc: the linq to sql layer for the database objects.

**Function Algorithms**

**Authenticate():** This method will check if the log information entered by the user is valid by comparing it to the database. The password entered by the user has to be hashed by SHA1.

* + 1. **MapService**



**Figure 30: MapService CD**

The MapService Class will handle actions required for the Map service. The zones class Javascript will call to this Service for dynamic queries to the database.

**Attributes**

Public DTMSDataContext dtmsdc: the linq to sql layer for the database objects.

**Function Algorithms**

**MapService():** Default Constructor, The function is empty.

**GetInformation():** Performs a query on the database for the information of the zone selected. And return the string in HTML format.

* + 1. **Masterpage**



**Figure 31: Masterpage CD**

The Masterpage Class will handle actions shared by all pages. The Masterpage will be the template webpage that all pages will follow. The template will keep the navigation menu for quick and easy navigation between pages. The navigation will be represented by web links to the page. The content area will then be replaced by the content of the page.

**Attributes**

Public DTMSDataContext dtmsdc: the linq to sql layer for the database objects.

**Function Algorithms**

**Page\_Load():** Page Load fills in all the user's information in the top right of the webpage and displays a link for the admin page for admins.

**OnLogout():** Abandon the user's session so that is won't persist for security and stability.

* + 1. **Messages**



**Figure 32: Message CD**

The Message Class will handle actions needed by the Message Center Page. This will include filtering and displaying messages addressed to the logged in user, along with bulletins addressed to all users. It will also include allowing the user to choose a message recipient, including a "Bulletinboard" option,\* fill in the message text, and send the message.

**Attributes**

Public DTMSDataContext dtmsdc: the linq to sql layer for the database objects.

Public DataView emailList: table of data to be displayed in the inbox list

Public DataView bulletinList: table of data to be displayed in the bulletin list

Private int BULLETIN: radio button value for recipient category

Private int PERSON: radio button value for recipient category

Private int ORGANIZATION: radio button value for recipient category

Private int ZONE: radio button value for recipient category

Private string BULLETINBOARD\_EMAIL: string to match BULLETINBOARD dummy account

Private string DELETED\_EMAIL: string to match DELETED dummy account

Private int SUCCESSPAGE: Success page value for Multiview

Private int ERRORPAGE: Error page value for Multiview

Private int ADMINISTATOR: administrator value for authorization

Private int OWNER: owner value for authorization

**Function Algorithms**

**Page\_Load():** This method handles the actions required when the Message page loads. It will set the recipient list based on query strings. It will populate the inbox list and bulletin list.

**updateInbox():** This method reloads the inbox list.

**updateBulletin():** This method reloads the bulletin list.

**inboxGridView\_OnRowDataBound():** This method adds a modal popup to all delete links on the inbox list.

**bulletinGridView\_OnRowDataBound():** This method adds a modal popup to all delete links on the bulletin list. This column will only exist if logged in user is admin or owner.

**inboxGridView\_PageIndeChanging():** This method changes the displayed inbox list based on pagination choice.

**gridView\_OnRowDataBound():** This method calls 2 sets of different functions: read and delete, for inbox and bulletin. The read commands display selected message in full into the inboxText field. The delete commands remove the message from the database. The delete commands have attributes added to them to create a modal confirmation before continuing.

**RecipientCategory\_OnSelectedIndexChanged():** This method changes the dropdown list based on recipient category chosen.

**recipient\_OnSelectedIndexChanged():** This method displays the recipients email If a single person is selected.

**Send\_OnClick():** This method sends messages or emails based on the choices made by the user. The outbox text is restricted to more than zero characters, and less than 2048.

* + 1. **NewEntry**



**Figure 33: NewEntry CD**

The NewEntry Class will handle actions required to create new Entries. The Entry will have pre-populated drop down list for easy selecting of fields. Once all the required fields are satisfied the entry can be submitted to the database.

**Attributes**

Public DTMSDataContext dtmsdc: the linq to sql layer for the database objects.

**Function Algorithms**

**Page\_Load():** Page load will populate all the drop downs on the page with the values in the database.

**Submit\_Button\_Click():** This method will create a new entry and submit it to the database using the information entered by the user. It's triggered when the user hits submit.

**DrugChange():**This method is called whenever the user selects a different value on the drug drop down. It will populate the drug alias drop down based on what drug was selected.

**OtherCheck():** This method is called whenever the user selects a different value on any of the drop downs except for the drug drop down. It is checking for "Other" to be selected.

**AddSideEffect\_Button\_Click():** This method is called when the user enters a new side effect and clicks "Add". It will make a new side effect as long as it doesn't already exist and add it to the database.

* + 1. **Report**



**Figure 34: Report CD**

The Report Page Class will handle actions needed by the Report Page. This consists primarily of generating Reports from the Entries in the database. The user will select the comparison field and the Report Page will generate a Line Chart, Bar Chart, Pie Chart or Heat map, graphing the Charts will all the Entries by the comparison field. The user has the additional options to filter which of the values in the comparison field to chart, and whether to filter which entries are included by any of the other fields.

**Attributes**

None

**Function Algorithms**

**Page\_Load():** This method handles the actions required when the Report page loads.

**line\_Link\_Click():** This method handles a user click on the Line Chart linkButton.

**bar\_Link\_Click():** This method handles a user click on the Bar Chart linkButton.

**pie\_Link\_Click():** This method handles a user click on the Pie Chart linkButton.

**heatmap\_link\_Click():** This method handles a user click on the Heatmap linkButton.

**comparelist\_SelectedIndexChanged():** This method handles a change in the selection of compare by field.

**compare\_Filter\_Gen():** This method prepares the checkboxlists used for additional filters.

**compare\_Filter\_list\_SelectedIndexChanged():** This method handles a change in the selection of compare by filters.

**generate\_Filter\_WhereClause():** This method generates the additional filter where clause and "Excluded" string from the checkboxlists.

**generate\_WhereClause():** This method generates the compare by field filtering where clause from the checkboxList

**printFriendlyVersion\_Click():** This method sets up the parameters and redirects the user to the print-friendly report page.

**sanitize():** This method encodes a string for use in generating reports

**generate\_ReportButton\_Click():** This method generates the report based on the parameters previously entered by the user.

**SelectEntry():** This method handles a click to "view entry" in the included entries gridView.

**filter\_Category\_CheckedChanged():**This method handles the user clicking a "master" additional filters checkbox

**showallComp\_Click():** This method handles the user clicking "Select All/Select None" for the compare by field filters.

**MainBox1\_CheckedChanged():** This method handles the user clicking a "master" additional filters checkbox

**FurtherContraintsButton\_Click():** This method handles the user clicking the "open further constraints" button

**FurtherContraintsClose\_Click():** This method handles the user clicking the "close further constraints" button

**PresetButtonOnClick():** This method handles the user clicking the "Generate Preset Report" button

**Prepare\_user\_zones():** This method selects all the zones belonging to the logged in user

* + 1. **ReportHelper**



**Figure 35: ReportHelper CD**

The ReportHelper Class will handle actions required to help the Reports Class.

This Class has functions to help with the report generation.

**Attributes**

Public DTMSDataContext dtmsdc: the linq to sql layer for the database objects.

Public int CORRECTION: Correction to use in colour generation

Public int NUMENTITIES: Number of searchable fields in an entry

Public int MONTHS\_CUTOFF: Number of months above which data is partitioned by year

Public double NORMVAL: Value to normalize "heats" to.

Public int LEGENDCUTOFF: Number of datapoints above which a bar or pie chart will require a legend

Public String [] entitiesList: This statically allocated array and the following one are the only part of this class that depends on the database structure.

Public IQueryable [] tablesList: This contains the entities required for joins, or null where no join is required

Public String [] color: List of colours to use for line charts

**Function Algorithms**

**normalize():** This method takes a list of ints and converts them to an array normalized from 0 to NORMVAL

**colourValues():** This method takes an array of ints and returns an array of hex colour strings representing the "heat" value of each number

**makeCheckArray():** This method takes a series of checkboxes and returns them as a list

**MakeCheckListArray():** This method takes a series of checkboxLists and returns them as a list

**generatePieChart():** This method generates a pie chart given the result of a query

**generateBarChart():** This method generates a bar chart given the result of a query

**generateLineChart():** This method generates a line chart given the result of a query

**createIncludedEntriesView():** This method creates the "Entries Included" gridView

**parseTotalMonths():** This method parses a date clause and returns the number of months between the start and end of the range.

* + 1. **ReportPrint**



**Figure 35: ReportPrint CD**

The ReportPrint Class will handle required to make the reports print able. This requires the page to have no style.

**Attributes**

None

**Function Algorithms**

**Page\_Load():** This method handles the actions required when the Print-Friendly Report page loads.

**generateHeatMap():** This method prepares the heat map parameters by querying initquery

**createHeatMap():** This method creates the script to render the heat map, given the array of zone colours

**SelectEntry():** This method handles a click to "view entry" in the included entries gridView.

**Render():** This method handles PageRender for the ReportPrint page.

**backButton\_Click():** This method sends the user back to the reports page when they click the "Back to reports" button

**desanitize():** This method decodes a sanitized string for use in generating reports

* + 1. **RequestMemebership**



**Figure 36: RequestMemebership CD**

The RequestMembership Class will handle actions needed by the Request Membership page. This will include creating a candidate member once the fields have been filled in.

**Attributes**

Public DTMSDataContext dtmsdc: the linq to sql layer for the database objects.

**Function Algorithms**

**Page\_Load():** Page load will populate the organization and zones drop downs.

**Submit\_Button\_Click():** This method will create a new user account with the values entered by the user.

**Organization\_DropDownList\_SelectedIndexChanged():** This method is called whenever a user changes a selected index on the organization dropdown. It just checks if "Other" was selected and displays the new organization text boxes.

**SendOwnerEmail():** This method sends an email to the owner notifying them that a user has requested membership.

* + 1. **ResetPassword**



**Figure 37: ResetPassword CD**

The ResetPassword Class will handle actions required to reset the contributors password. The new temporary password will be sent to the contributors email account.

**Attributes**

Public DTMSDataContext dtmsdc: the linq to sql layer for the database objects.

**Function Algorithms**

**SubmitButton\_Click():** This method checks if the user exists, reset's the user's password and sends the user an email with the new password.

**SendNewPasswordEmail():**This method will send an email to the user with their new password.

* + 1. **ViewContributors**



**Figure 38: ViewContributors CD**

The ViewContributors Class will handle the actions required to view a Contributor’s information.

**Attributes**

Public DTMSDataContext dtmsdc: the linq to sql layer for the database objects.

**Function Algorithms**

**Page\_Load():** Page Load will display all the specified contributor's information on the page.

**Zone\_List\_Click():** This Event Handler is called when the user clicks on the link for any zone on the page. It will redirect the user to the zones page for the zone that was clicked.

* + 1. **ViewEntry**



**Figure 39: ViewEntry CD**

The ViewEntry Class will handle the actions required to view an Entry’s information.

**Attributes**

Public DTMSDataContext dtmsdc: the linq to sql layer for the database objects.

**Function Algorithms**

**Page\_Load():** Page Load will display all the information about the specified entry. If none is specified then display the last one entered.

* + 1. **ViewOrganization**



**Figure 40: ViewOrganization CD**

The ViewOrganization Class will handle the actions required to view an Organization’s information.

**Attributes**

Public DTMSDataContext dtmsdc: the linq to sql layer for the database objects.

**Function Algorithms**

**Page\_Load():**  Page load will display all the information about the specified Organization. If none is specified in the query string than a redirect occurs and the first organization is displayed.

* + 1. **Welcome**



**Figure 41: Welcome CD**

The Welcome Class will handle actions required by the Welcome page. This will include loading the 10 most recent entries and bulletin board.

**Attributes**

Public DTMSDataContext dtmsdc: the linq to sql layer for the database objects.

**Function Algorithms**

**Page\_Load():** This function will databind the 10 most recent entries and the 10 most recent bulletinboard entries. The data will be bound to a gridview on the front page.

* + 1. **Zones**



**Figure 42: Zones CD**

The Zones Class will handle actions needed by the Zones Page. This will include loading the map of Zones and associating each section of the map with a Zone from the database. When the user clicks one of these sections of the map, the Zones Class will load the Contributors and the most recent Entries for that Zone from the database for display.

**Attributes**

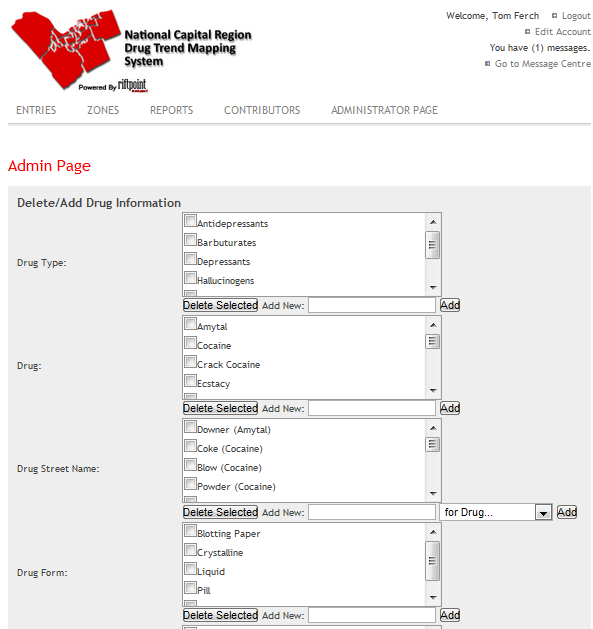
Public DTMSDataContext dtmsdc: the linq to sql layer for the database objects.

**Function Algorithms**

**Page\_Load():** The page load will only execute if there is a query string in the request field. The Request string will be parsed and send a request to the map service.

**Render():** This method handles the actions required when the zones page has been rendered. Why all the code was put into the render was because the map\_canvas div tag needed to be created before the Javascipt was created.

* 1. **Software Interface Descriptor**
     1. **Administrator**

****

**Figure 43: Admin Page HCI**

**Narrative:**

The Admin Page shows the Administrator the list of items in each Entry Field. It allows the Admin to add new items or delete existing items in any of the tables. When a Drug or Street Name is deleted, all Entries containing those items are deleted. When another item is deleted, that field will be changed to null in all the entries containing it. This page is only available to Administrators and Owners.

**Controls:**

- The following fields can be altered:

- Drug Type

- Drug

- Drug Street Name

- Drug Form

- Drug Colour

- Pill Stamp

- Pill Shape

- Side Effects

- Organization

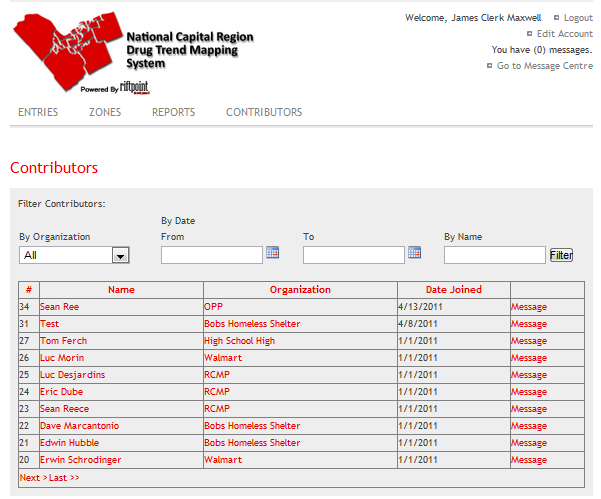
- Each field can be altered by an administrator or owner by either adding or deleting items in the corresponding list next the field name.

- Each item in the list can be selected to be deleted with the corresponding delete button for that field.

- New items can be added to a field by typing the name and clicking the add button for the corresponding field.

- Exception: New Drug Street Names must also include Drug name associated to it by using the dropdown menu next to the add button.

* + 1. **Contributors**



**Figure 44: Contributor HCI**

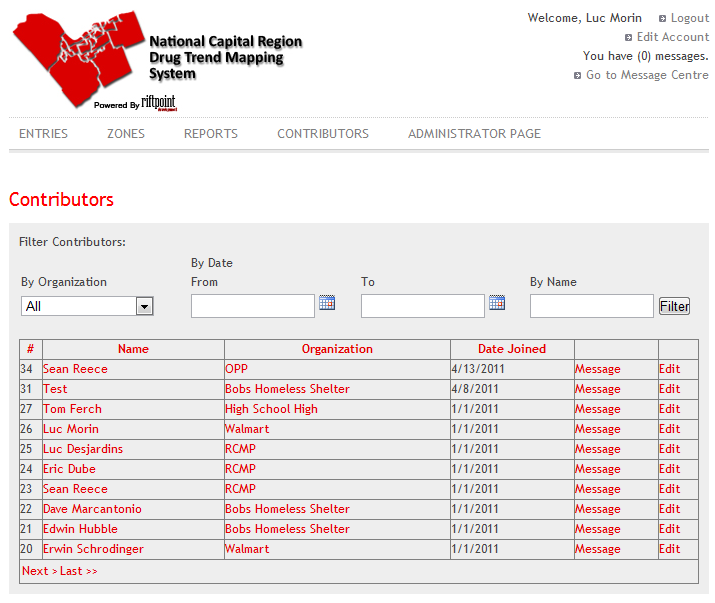
**Narrative:**

The Contributors Page shows the user the list of Contributors to the NCRDTMS in a DataGrid. It allows the data to be sorted by any of the fields. It also allows any Contributor in the Grid to be clicked to view all of that contributor’s information. If you select the contributor’s organization the page will be redirected to a view organization page. The message link will redirect the user to the Message Center. Buttons appear for the administrator to edit Contributors' account details.

**Controls:**

* Each Contributor's name is a link to the View Contributor page, which will display all the contributors’ information.
* Each Organization can be selected to view that contributor’s organization information.
* Message link will redirect the user to the message center.
* Each Header on the Datagrid is a button to change the sort order of the list of Contributors. This includes Name, Organization, Date Joined. Each can be clicked once to sort by that field, and again to reverse sort.

* + 1. **Contributors (When Logged in as Admin)**

**Figure 45: Contributors Administrator HCI**

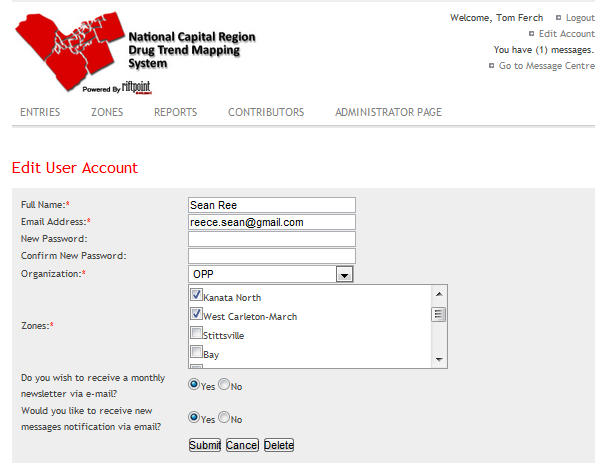
**Narrative:**

The following Contributors represent what an administrator view of the Contributors page. The View will Contributor page will change if an administrator is logged in.

**Controls:**

* An Edit button exists to the right of each record in the datagrid. It is enabled and displayed only to users logged in as administrators.
* This button redirects the administrator to the edit contributor page.

* + 1. **Edit Account**

****

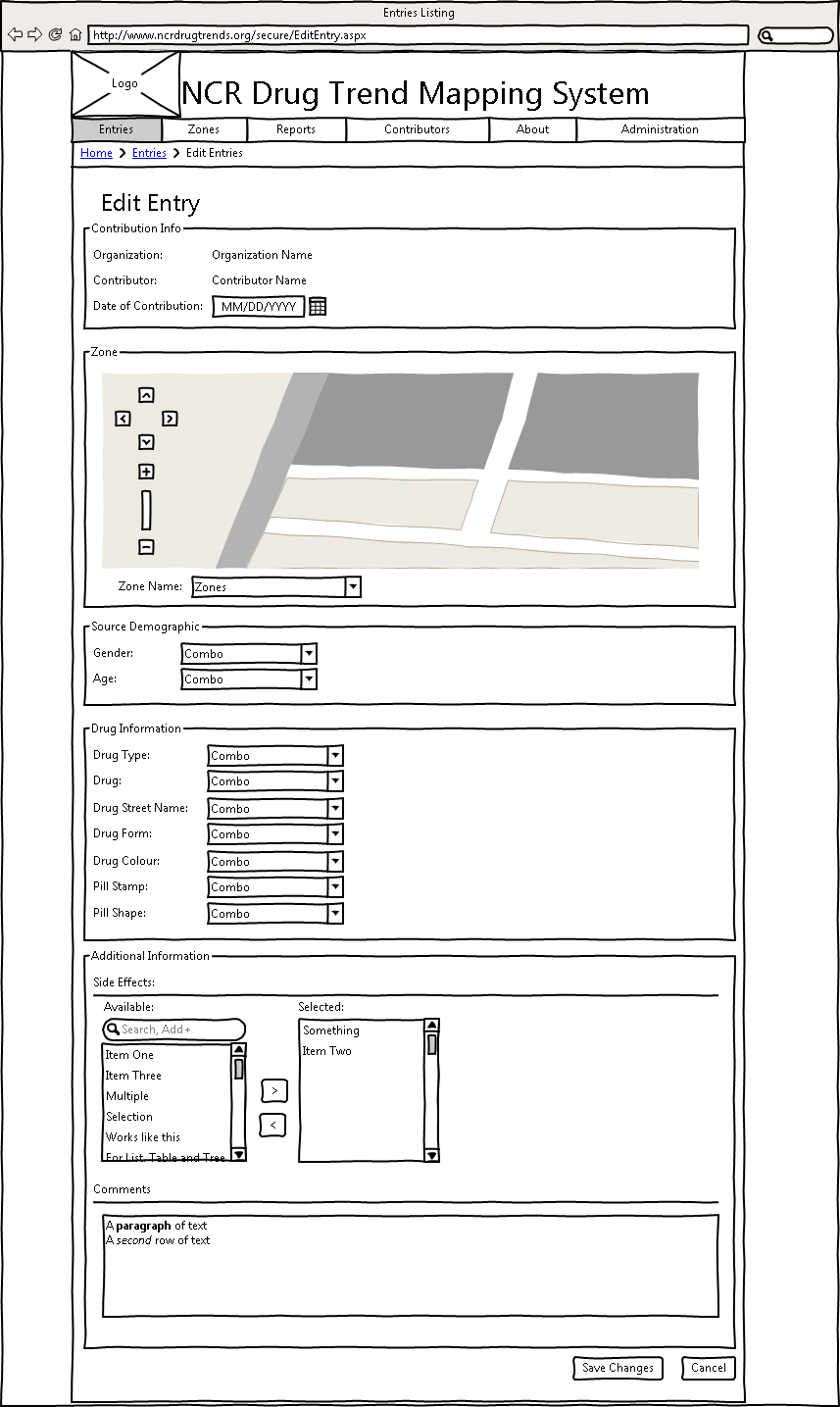
**Figure 46: Edit Account HCI**

**Narrative:**

The Edit Account Page allows an Administrator to edit the details of a particular user account. This includes adding or subtracting zone associations, giving a password reset, changing their active email address, name, and organization. The controls will be initialized with the current values of the Account in the database, except the password boxes, which are left empty.

**Controls:**

* Zone Checkboxes: A checkbox is displayed for each Zone in the System. Each box that is checked indicates a zone that the user being edited is associated with.
* Password: A textbox to enter a new password for this user if desired.
* Confirm Password: A second textbox to enter the new password for the user. If the text does not match that in the first box, an error will be generated.
* Email Address: A textbox to enter a new email address for the user.
* Full Name: A textbox to enter the name of the user.
* Organization: A drop down to enter the name of the Organization the Contributor works for.
* Submit Changes: A button which submits any changes made to the user account to the database.
* Cancel Changes: A button which cancels the changes made.
* Delete: Will delete the account in the database.
  + 1. **Edit Entry**



**Figure 47: Edit Entry HCI**

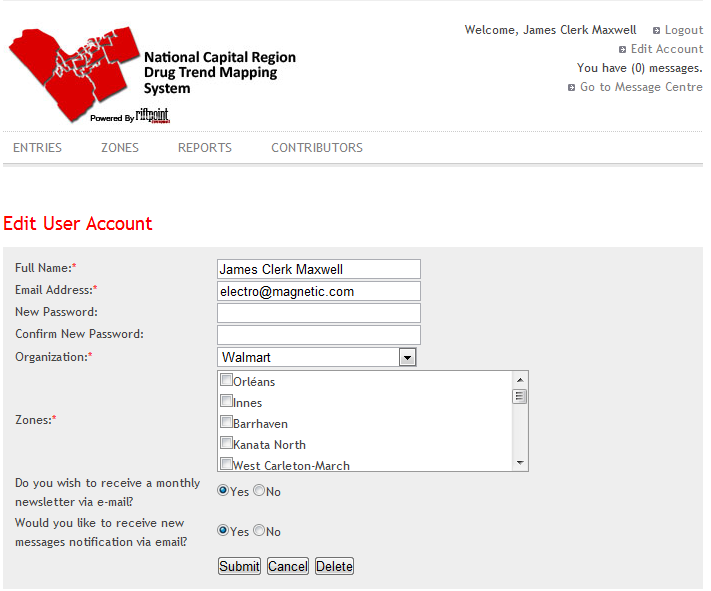
**Narrative:**

The Edit Entry Page allows an Administrator to edit the details of a particular Event Entry. This includes changing the date, zone, source demographic gender and age range, drug Information, drug type, drug, drug street name, drug form, drug colour, Pill Stamp and Pill Shape. The optional drug Information is side Effects and the comments. The controls will be initialized with the current values of the Entry in the database.

**Controls:**

* Date Gathered Calendar: A calendar based text field will pop up with a calendar for easy date selecting.
* Zone map: A map that shows where the selected zone lies.
* Zone Dropdown: Dropdown menu to select the zone.
* Gender Dropdown: Dropdown menu to select the gender of the drug user.
* Age Range Dropdown: Dropdown menu to select the approximate age of the drug user.
* Drug Type Dropdown: Dropdown menu to select the type of drug reported.
* Drug Dropdown: Dropdown menu to select the drug reported.
* Drug Street Name Dropdown: Each Drug has drug street names associated to it. This list will change depending on the drug selected.
* Drug Form Dropdown: the format of the drug used.
* Drug Colour Dropdown: Drop down of the colour of the drug.
* Pill Stamp Dropdown: Dropdown menu to select the stamp of the pill reported. It appears only if Pill is selected as the drug form.
* Pill Shape Dropdown: Dropdown menu to select the pill shape reported. It appears only if the Pill is select as the drug form.
* All Fields will have an “Other” selection will then show a text box to add the new entry into the list of the dropdown.
* Additional Information,
* Side Effects Selector: a drug can have many side Effects associated to its entry.
* Comments Textfield: A textfield to enter the comments to be attached to the entry.
* Submit Entry Button: Button to create the entry and submit it to the database.
* Cancel Entry Button: Button to cancel all changes made to the current entry information.

* + 1. **Edit Settings**

****

**Figure 48: Edit Settings HCI**

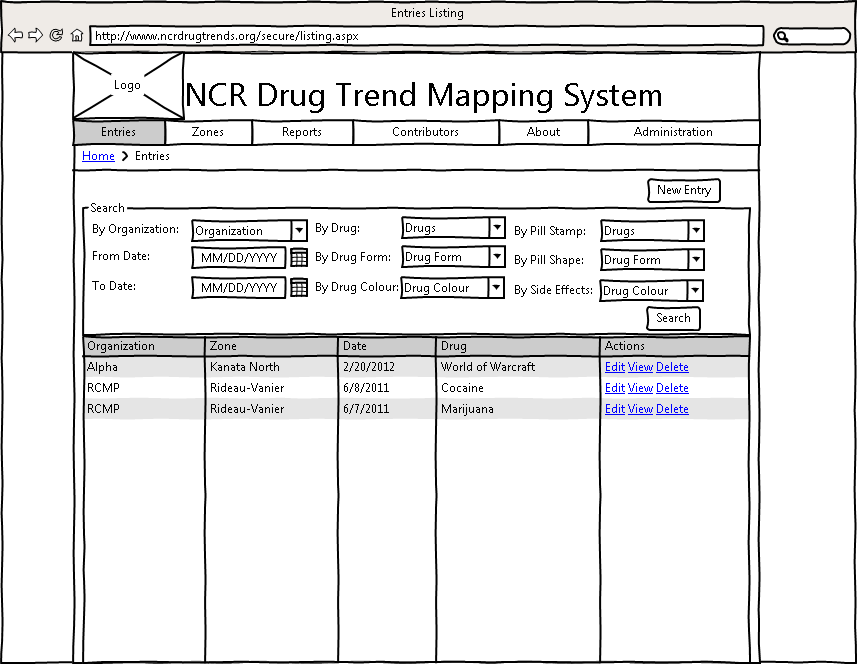
**Narrative:**

The Edit Account Page allows a User to edit the details of their own user account. This includes changing their Full Name, Email Address, password, organization or zones. They can also change the subscribing or unsubscribing from the monthly email newsletter. The controls will be initialized with the current values of the Account in the database, except the password boxes, which are left empty.

**Controls:**

* Full Name: A textbox to enter the name of the user.
* Email Address: A textbox to enter a new email address for the user.
* Change Password: A textbox to enter a new password for this user if desired.
* Confirm Password: A second textbox to enter the new password for the user. If the text does not match that in the first box, an error will be generated.
* Organization: A textbox to enter the name of the Organization the Contributor works for.
* Zones: a selection box that the user is associated to. The contributor can be associated to multiple zones.
* Newsletter Checkboxes: A pair of checkboxes to select whether to receive the newsletter or not.
* Email Checkboxes: A checkbox to select if the user would like to receive their message via email.
* Submit Changes: A button which submits any changes made to the user account to the database.
* Cancel Changes: A button which resets the text in each box to the value currently stored in the database. This will clear all changes made.
* Delete Changes: A button which will delete the users account upon a confirmation dialog box.

* + 1. **Entries**



**Figure 49: Entries HCI**

**Narrative:**

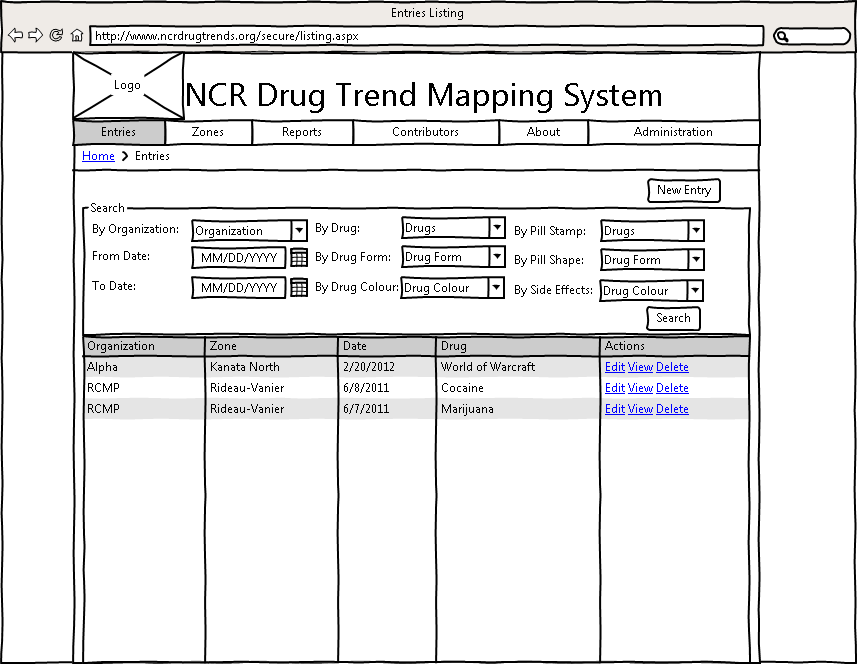
The Entries Page shows the user the list of Event Entries to the NCRDTMS in a DataGrid. It allows the data to be sorted by any of the fields by selecting the titles at the top of the grid. There are also filters that you can filter by, Calendar: from date, Calendar: to date, Textbox: by the entry number, Textbox: by the contributors name,

Dropdown for the organization, Dropdown for the zones, Dropdown for the drug and finally a filter button.

**Controls:**

* Create Entry Button: Button that redirects the User to the Create Entry page.
* Calendar From date: Calendar based field to select the from date.
* Calendar To date: Calendar based field to select the to date.
* Textbox Entry Number: Searching a specific entry by its ID.
* Textbox by Name: Search all the entries for entries submitted by a person with that name.
* Dropdown for Organization: to search all entries for a specific organization.
* Dropdown for Zone: filter all entries by a specific zone from a list.
* Dropdown for Drug: filter all the entries by a specific drug.
* Button Filter: Filter by the specific characteristics specified by the fields above.
* Each Header on the Datagrid is a button to change the sort order of the list of Event Entries. This includes Poster, Organization, Zone, Drug, and Date. The header can be clicked to sort by the field, and again to reverse sort.

* + 1. **Entries (When Logges as Admin)**



**Figure 50: Entries Administrator HCI**

**Narrative:**

The administrator has greater functionality then a regular user.

**Controls:**

* An Edit button, view button, and delete button exists to the right of each record in the datagrid. It is enabled and displayed only to users logged in as administrators. This button redirects the administrator to the edit event entry page.

* + 1. **Login**

****

**Figure 51: Login HCI**

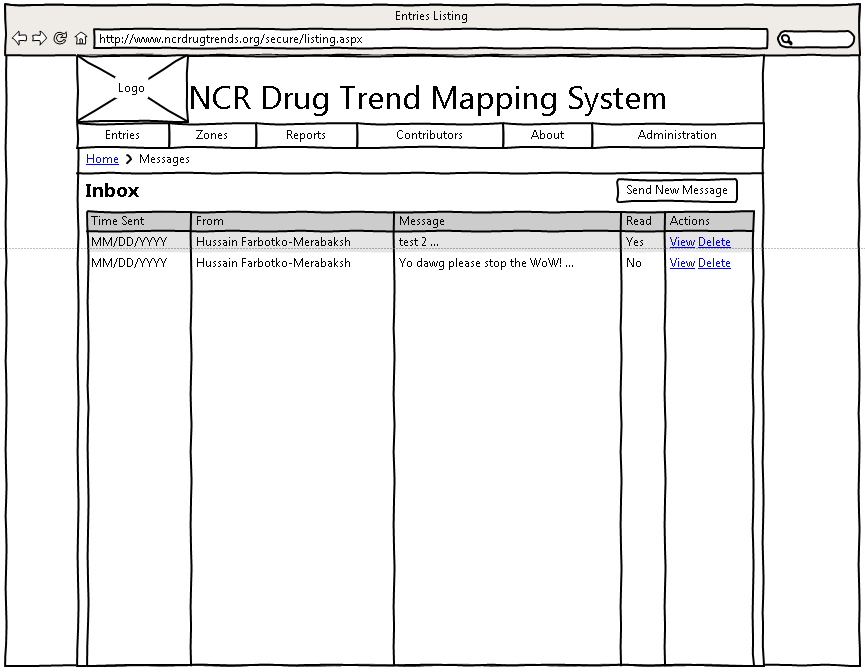
**Narrative:**

The Login Page allows the User to log on to the system or request membership. It also displays a description of the purpose of the NCRDTMS.

**Controls:**

* Email Textbox: Textbox to enter the email address of your user account.
* Password Textbox: Textbox to enter the password of your user account.
* Request Membership Link: Link to the Request Membership page.

* + 1. **Message Centre**



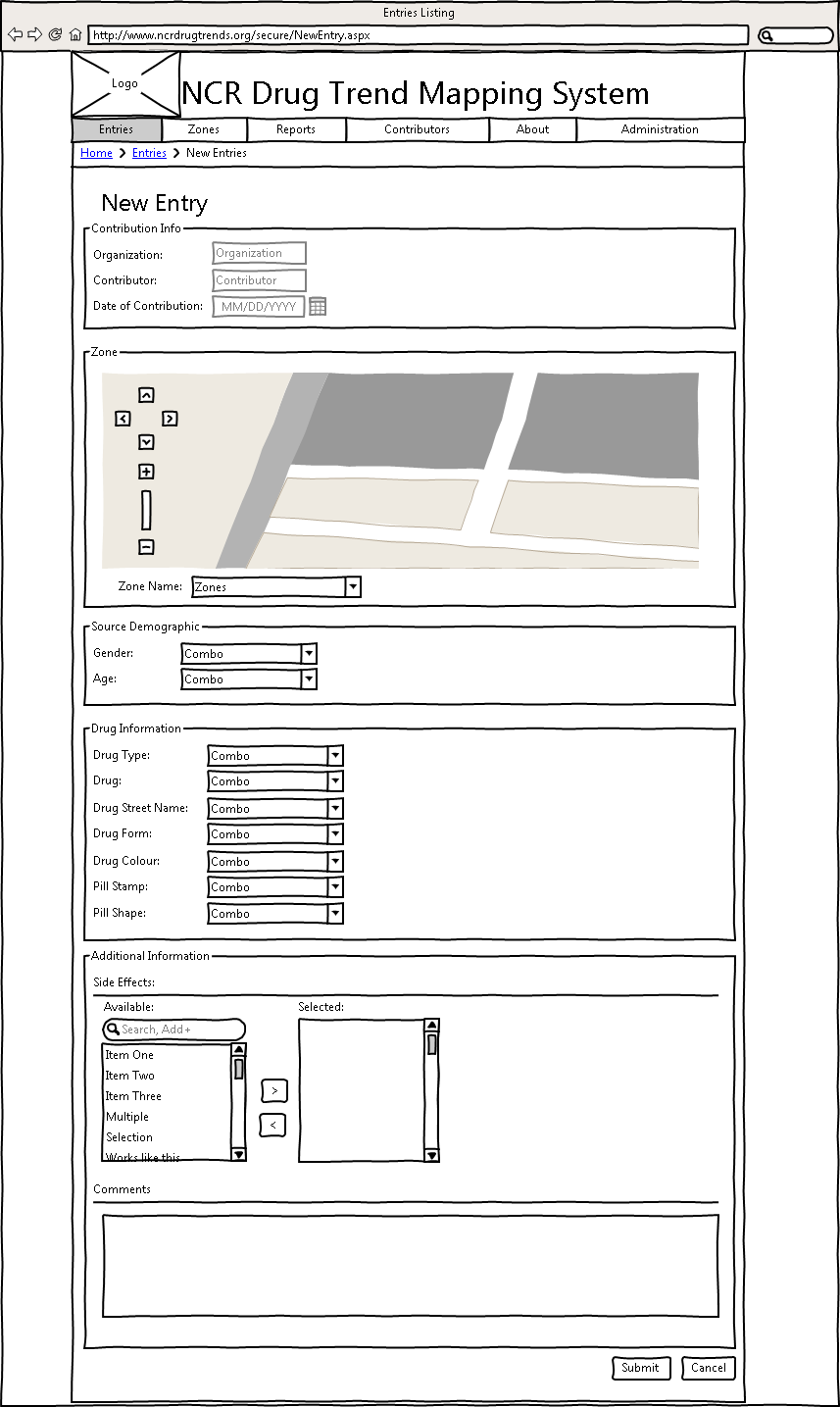
**Figure 52: Message Centre HCI**

**Narrative:**

The Message Center allows the User to view the messages addressed to them or to all users. It also allows the User to compose a message and send it to another User, Bulletin, Organization or by Zone.

**Controls:**

* Inbox List: A selectable list which displays the sender, first line, and date sent of each message addressed to the current user. When a message is selected in the list, its full text will appear in the Message Output text below.
* Bulletinboard List: Selectable list of bulletins board message.
* Message OutputText: An OutputText which will display the full text of the message that is selected in the inbox list.
* Send New Message Button: Navigate to the send new message page.
  + 1. **New Entry**



**Figure 53: New Entry HCI**

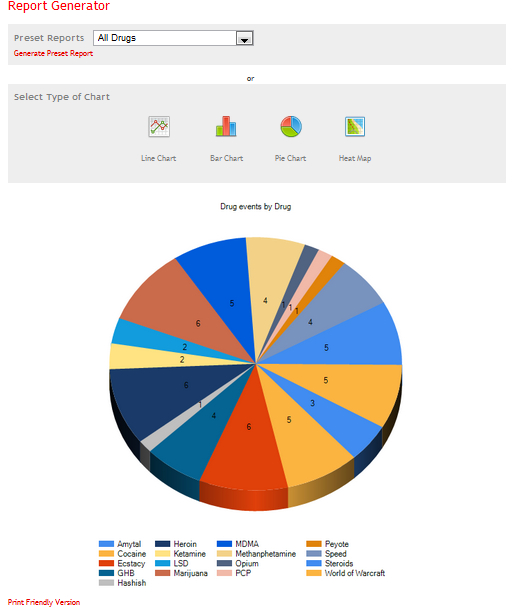
**Narrative:**

The New Entry Page allows a User to enter a new Event Entry. This includes entering the date, zone of event, zone of purchase, source demographic gender and age range data, drug type, drug, drug alias, drug effects, side effects, and the applicable physical description fields.

**Controls:**

* Calendar Date: Select a date for the event.
* Zone Map: Shows the location of the specified zone
* Zone Applicable Dropdown: Dropdown menu to select the zone of the event. If the user is associated to 1 or more zones he will half to select a zone.
* Gender Dropdown: Dropdown menu to select the gender of the drug user.
* Age Range Dropdown: Dropdown menu to select the approximate age of the drug user.
* Drug Type Dropdown: Dropdown menu to select the type of drug reported.
* Drug Dropdown: Dropdown menu to select the drug reported.
* Add Category - Drug Link: Link to add a new drug to the list of possibilities.
* Drug Street name Dropdown: Dropdown menu to select the drug street name used once a drug was selected a drug street name is associated to it.
* Drug Form Dropdown: dropdown menu to select the drugs form factor, Pill, Liquid etc.
* Drug Colour Dropdown: dropdown menu to select the drug colour.
* Pill stamp Dropdown: if the drug form is a pill then this field will be selectable.
* Pill shape Dropdown: If the drug form is a pill then this field will be selectable.
* Notable side effect Select menu: Any side effects that were reported during the drug usage.
* Comments Textbox: Comments that the Contributor wishes to add to the entry.
* Submit entry Button: Submit the entry to the database, If the form has all the required fields the entry will be submitted to the database.
* Cancel entry Button: Cancel the entry being submitted to the database.

* + 1. **Report**

****

**Figure 54: Report HCI**

**Narrative:**

The Report Page allows the user to generate reports based on the data stored in

the NCRDTMS database. It allows a report to be created based on a primary

comparison field selected by the user. The user may also filter the entries to

be used in composing the report by any of the fields in an Entry. The Report

Page also allows the user to load one of a set of pregenerated common report

parameters for ease of use. The page will include a small blurb explaining to

the user how to print the report or save it locally, along with a warning that

the data is not scientific.

**Controls:**

* Pre-generated Report Dropdown: a dropdown which generates and displays a report based on the report selected by the user.
* Chart List: A list which allows the user to choose a chart type.
* Primary Comparison Dropdown: A dropdown will allow the user to select a primary comparison field.
* More controls will exist to allow the user to select filters and additional options based on the chart type selected.
* A generate button will generate and display the chart when clicked

* + 1. **Request Membership**

****

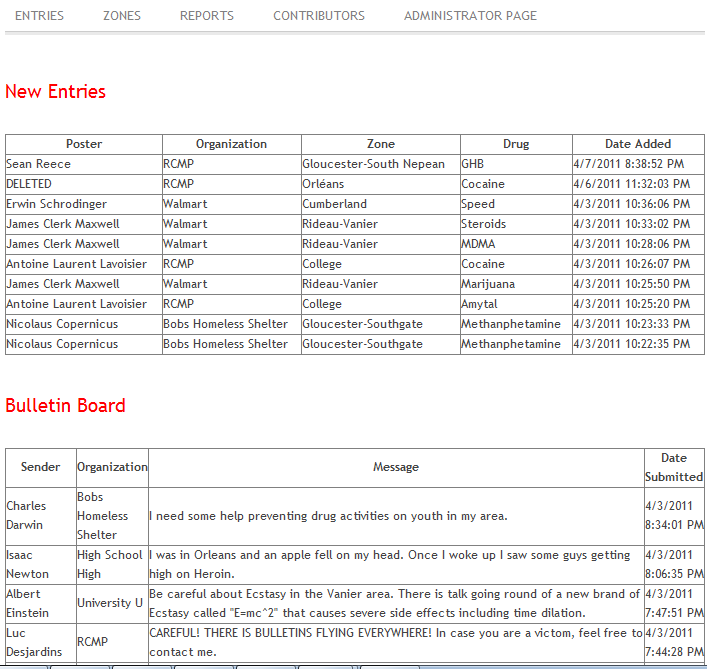
**Figure 55: Request Membership HCI**

**Narrative:**

The Request Membership Page allows a Potential User to enter the details of the user account they would like to create. This includes entering the email address, name, and organization, and subscribing or unsubscribing from the monthly email newsletter. The potential user can also fill in a field of comments that will be sent to the Owner with the account membership request.

**Controls:**

* Full Name: A textbox to enter the name of the user.
* Email Address: A textbox to enter the email address for the user.
* Password Textbox: Password textbox field, password associated to the account.
* Organization: A dropdown to enter the name of the Organization the Potential Contributor works for.
* Newsletter Checkboxes: A pair of checkboxes to select whether to receive the newsletter or not.
* Comments Field: A text area to enter any comments to be sent to the Owner with the membership request.
* Submit Request: A button which submits the requested user for the owner to approve or reject.
  + 1. **Welcome**

****

**Figure 56: Welcome HCI**

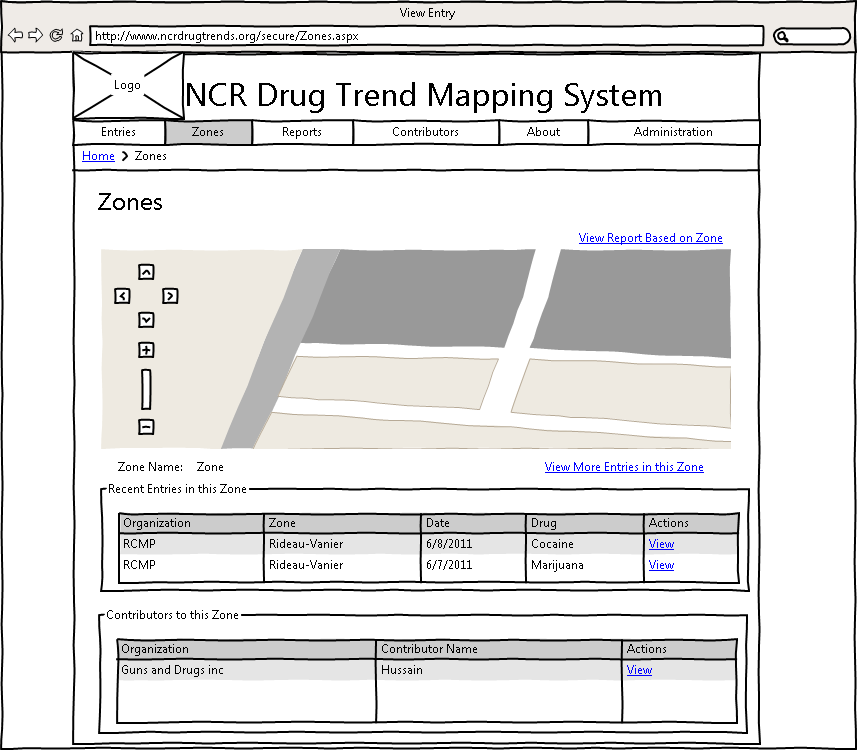
**Narrative**

The Welcome Page shows the user the list of Event Entries to the NCRDTMS and Messages addressed to them or to all posted since their last logon in a pair of DataGrids. It allows the Entries to be sorted by any of the fields. The messages are shown sorted by date.

**Controls:**

* new entries Table: The 10 most recent entries to the system. The entries will display the poster, Organization, Zone, Drug and the date it was submitted.
* Bulletin board Table: The 10 most recent bulletins sent to all the contributors.

* + 1. **Zones**



**Figure 57: Zones HCI**

**Narrative:**

The Zones Page shows the user the list of Zones on the NCRDTMS on a map. Each section of the map can be clicked to show the Contributors and Recent Entries to the Zone in a pair of DataGrids. Each allows the data to be sorted by any of the fields. The Contributors DataGrid also allows any Contributor in the Grid to be clicked to redirect to the Message Center. The Mapping system used is Google maps.

**Controls:**

* Each Contributor's name is a link to the Message Center with that Contributor's name as a parameter.
* Each Header on the Contributors Datagrid is a button to change the sort order of the list of Contributors. This includes Name, Organization, Zones, and Date Joined. Each can be clicked once to sort by that field, and again to reverse sort.
* Each Header on the Entries Datagrid is a button to change the sort order of the list of Entries. This includes Poster, Organization, Zone, Drug, and Date Added. Each can be clicked once to sort by that field, and again to reverse sort.
* View Report Based on Zone: Go to the reports page with a report on the currently selected zone.
* View More Entries in this Zone: View a more detailed listing of entries in specified zone.

1. **Restrictions Limitations and Constraints**

The Environment being used is a Windows Server R2 2008 with SQL Server 2008.