Final Report

Rate my world leader

Adam Finer, Thomas Bonatti, John Hamilton

2014-2015

**Table of Contents**

**Table of Contents1**

**1 Executive Summary2**

**2 Introduction2**

**3 High Level Problem Summary2**

3.1 Elevator Statement2

3.2 Summary of primary success criteria3

3.3 Scope3

**4 Detailed Problem Statement3**

4.1 Function3

4.2 Form3

4.2.1 Availability3

4.2.2 Usability4

4.2.3 Performance4

4.2.4 Security4

4.2.5 Maintainability4

4.3 Economy4

4.4 Time4

**5 Key Stakeholders4**

**6 References5**

**7 Appendix5**

**8 Index5**

**9 Glossary5**

**EXECUTIVE SUMMARY**

This final report document begins with a description of the initial problem that this project was intended to solve. The solution to that problem, a discussion of the key challenges that were faced over the course of the project, and a description of the database design will follow. The description of the database design will include the integrity constraints that were applied, the indexing scheme, views and stored procedures that were used, and a discussion of the pros and cons of the design.

**INTRODUCTION**

This document is the final report on the Rate My World Leader project developed by a team consisting of Adam Finer, Thomas Bonatti, and John Hamilton. The purpose of this document is to properly and effectively describe the problem described in previous documents, the Final Problem Statement, and to additionally describe the solution implemented by the team.

**Problem Description**

In the modern world, there are numerous people from hundreds of different areas who are considered to be world leaders. There are presidents, prime ministers, kings, queens, senators, governors, and numerous other positions that those leaders fill. Taking the concept of the Rate My Professor website of rating professors on a scale and commenting on them, the team envisioned an idea to create an application that would allow users from around the world to rate the leaders of the world. The end result of this program was intended to be a user interface called Rate my World Leader on which a user could login and use the implemented features of the application.

Below is the list of features for the final version of the database, there are some additional features that have been added on since the final problem statement was turned in.

|  |  |
| --- | --- |
| Feature Name | Feature Explanation |
| User accessible application | The way to access the application is through a downloadable application |
| Administrative privileges | If a user is of administrative status, he or she can delete non admin users, add and update the status of world leaders, upgrade non-admins to admin status, and many other actions that normal users can not perform. |
| Users | Any person can access the application and create an account to access the application interface |
| Rate leaders | Any user, once they are logged into the database, can add a single rating for any leader. The rating consists of a scalar from 0-10 where 0 means they hate the leader and 10 means they believe that leader is the best leader on the planet and a description of the leader. |
| *Game of Thrones* References | We reference characters and places from the *Game of Thrones* universe |
| Leaders | A list of some Major World leaders is include so that the current user database (just the three developers and some dummy users) can rate leaders |
| List of Locations | To allow for possible future development, a list of locations including all of the Continents, Countries, all of the US states, and a few cities were created. As of now, not all of those places have leaders (only countries do due to the scope the team decided to go with). |
| Political Parties | A list of political parties for leaders is included for additional information about world leaders |

**Solution Description**

The solution that the team decided upon was to develop an application using java graphics and Microsoft SQL server Management Studio 2012 to store the information related to the database.

## Front End

## back End

The back end of the solution, which was created using the SQL server software, contains the data the front end GUI uses. This data base contains the seven tables: Leader, Leads, Location, PartyMember, PoliticalParty, Rating, and User\_Account which interact with the user interface to enhance the user experience of creating and browsing world leaders. This database currently exists on the server titan.csse.rose-hulman.edu provided by Rose Hulman. Once this server is shut down though, it is still unknown where the database will be stored.

**Key Challenges**

Challenge: Little experience using SQL Server

Solution: For most problems, syntax could be looked up online, because of the extensive website provided by Microsoft, or in the lecture notes from class.

Analysis: The internet was useful for answering all of the questions.

**Database Design**

## Integrity Constraints

## Stored Procedures

|  |  |
| --- | --- |
| Stored Procedure name | Procedure Purpose |
| Add Rating | Allows the user to properly add a rating on any leader of their choosing |
| adminaccount | Allows an admin to make any normal user into an administrator |
| changepassword | Allows an user to change their password as long as they have their old password |
| CreateLogin | Allows someone to create a non-admin user account with a valid email, username, and password. Through this procedure, the password given is Hashed and saved in the database. |
| Delete\_account | Allows a normal user to delete their account which also deletes any ratings they may have created. |
| loginsp | Allows a registered user to log into the application. |
| RemoveAdminPrivlage | Allows an administrator to remove admin privileges from another administrator. |
| RemoveRating | Removes the specified rating, given by the primary key of Rating. |
| usersratings | Creates a table of all the ratings for a given user. |

## Views

The Rate Your World Leader database contains the following six views:

* dbo.AllLeaderRatings
* dbo.AverageLeaderRating
* dbo.CountryView
* dbo.PartyMembers
* dbo.PublicUserView
* dbo.UserRatings

These views were used to monitor certain features of the database during its creation. For example, the CountryView was used to see what countries had already been added to the database so that we did not add any repeats. Another use for the views was to create viewing pages for the application. For example, the UserRatings view can be seen from a window in the application. Generally, views were not necessarily needed for this particular project, because most of tables seen in the application are created by select statements and stored procedures.

## Indexes

There was only one developer created index. A nonclustered index was created for the Ratings table called useraccount which was created because there are searches done on the database by user for searching for ratings by username. Clustered indexes were pre-created by Microsoft SQL Server Management Studio on all tables based on their primary keys. If this were not the case, clustered indexes would have been created on the USER\_ACCOUNT table based on username, the LEADER table based on leader\_id, and on the LOCATION table based on the ppID for each location in the table.

## Triggers

The following triggers are listed below with explanations for why they are in the database.

1. insertLeadsTrigger- the intended purpose of this trigger was to prevent any leader from leading the same location at the same time as another leader and to prevent a leader from leading two places at once at the same time.
2. insertPartyMemberTrigger- Similar to the previous trigger, this trigger was intended to prevent, on insertion, for a leader to belong to a single party at the same time that he or she belongs to another given party.
3. CreateRating- This trigger was intended to properly insert a rating in such a fashion that the time stamp is correct for the rating. Also, the trigger increments whatever user created the rating’s number\_rating attribute in the USER\_ACCOUNT tables by 1.
4. DeleteRating- this trigger does the opposite of the previously defined trigger, accept on delete instead of insert
5. DeleteAccount- this trigger deletes the tuple specified in the deletion statement as well as deletes that user’s ratings from the Ratings table to keep integrity constraints sound.

# **Design Analysis**

## Strengths

* User passwords in the database are hashed with the username of what ever user that password belongs to allowing for enhanced security of user passwords

## Weaknesses

* Users are unable to view their passwords.
* The team did not

**Appendix**

## Relational Schema

**Leader**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Leader\_id | gender | bday | fname | lname | midint |

**Leads**

|  |  |  |  |
| --- | --- | --- | --- |
| leader\_id | startdate | location\_id | enddate |

**Location**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ppID | superppID | type | name | population |

**PartyMember**

|  |  |  |  |
| --- | --- | --- | --- |
| leader\_id | startdate | party\_id | enddate |

**PoliticalParty**

|  |  |  |
| --- | --- | --- |
| ID | placeOfOrigin | party\_name |

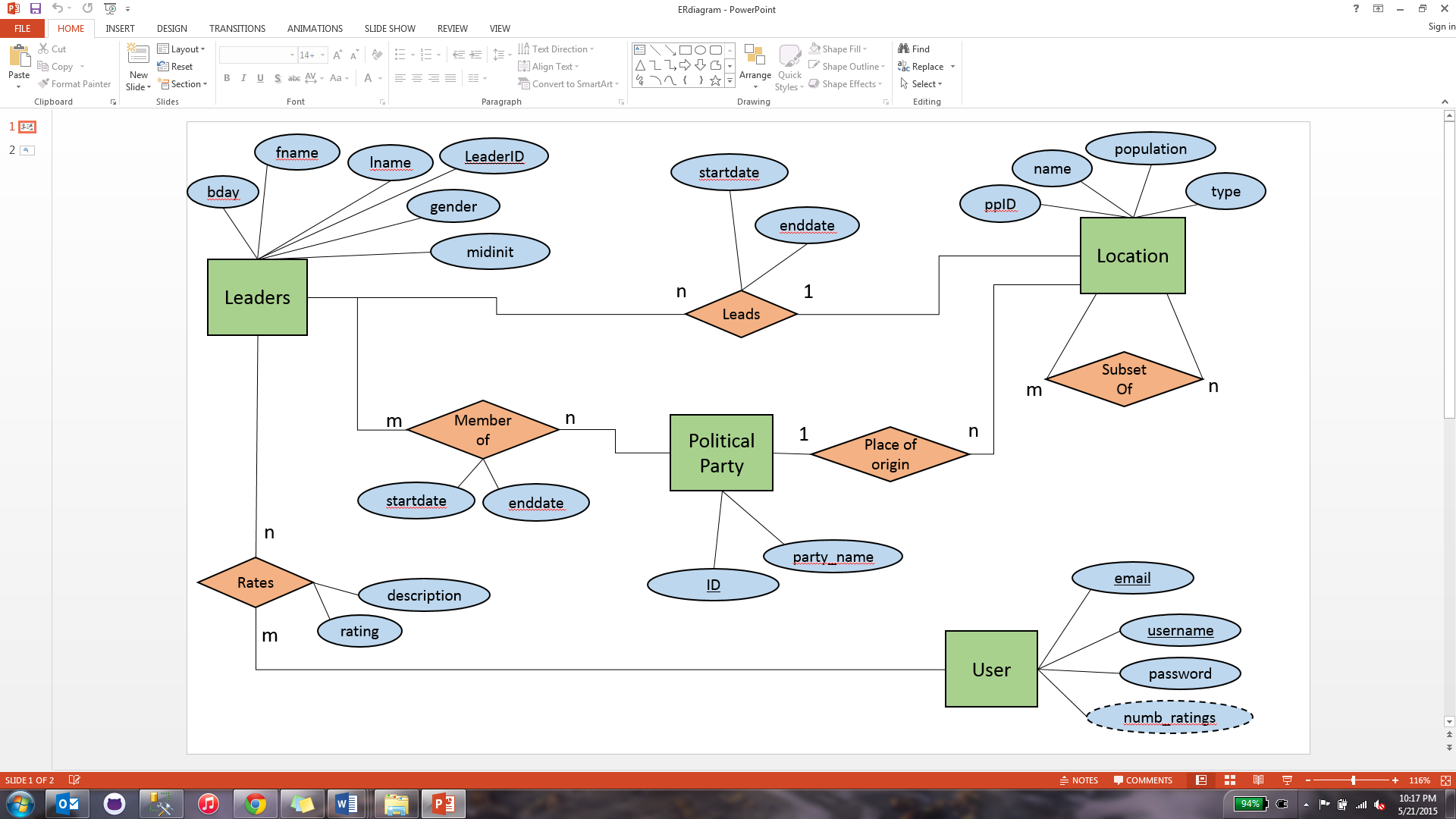
**Rating**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| leader | username | rating | text | date |

**UserAccount**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| username | email | number\_ratings | passwordHash | isAdmin |

## Entity RelationShip Diagram



**Glossary**

**References**

"RateMyProfessors.com – Find and Rate Your Professor or Campus." Rate My Professors. Web. 3 Apr. 2015. <http://www.ratemyprofessors.com/>.

*Central Intelligence Agency*. Central Intelligence Agency, n.d. Web. <https://www.cia.gov/library/publications/the-world-factbook/>

*Wikipedia*. Wikimedia Foundation, n.d. Web. 21 May 2015. <https://www.wikipedia.org/>

Game of Thrones TV show and Book series by George R. R. Martin

**INDEX**