Universal Planner

Final Report

Analysis of the Project

Ishank Tandon

Jared Hoffman

5/22/2015

Git Repository: <https://github.com/tandoni/CSSE333Project>

Git Username: tandoni

Contents

[Executive Summary 3](#_Toc420096910)

[Introduction 4](#_Toc420096911)

[Problem Description 5](#_Toc420096912)

[Features 5](#_Toc420096913)

[Solution Description 6](#_Toc420096914)

[Front-End Description 6](#_Toc420096915)

[Back-End Description 6](#_Toc420096916)

[Appendix 7](#_Toc420096917)

[Relational Schema 7](#_Toc420096918)

[Entity-Relationship Diagram 8](#_Toc420096919)

[Explanation of Entity-Relationship Diagram 9](#_Toc420096920)

[Glossary 10](#_Toc420096921)

[Index 11](#_Toc420096922)

# Executive Summary

This final report documents begins with describing the initial problem which led us to make this application. The solution application addresses the problem, gives the details of the overall database design, provides an analysis of all the key features, and also provides a security analysis of the solution. An entity-relationship diagram can be found at the bottom. Alongside, a relational schema is also attached. Furthermore, document also has a glossary of key terms, and an index.

# Introduction

This document is the final report of our project which is called ‘Universal Planner’ developed by Ishank Tandon and Jared Hoffman. This document provides the final proposed solution and some new features which were added after the last report which was turned in. This document also includes the Entity-Relationship Diagram and the Relational Schema. A detailed description and analysis of the database can also be found.

# Problem Description

Applications are designed and made to help people and reduce the amount of work which they have to do manually. For example, an organization without any digital support can have numerous problems maintaining information. Our application, Universal Planner, is a digital planner which was designed to help people and organizations to create, remove, host, attend, and edit events. To further the success of the application, we made sure that users are able to see the location of the events on a Map, therefore we integrated Google Maps for the same. The goal of the team was to establish an intuitive, well designed, and a responsive web application and make it available to organizations and general public free of charge.

## Features

1. Web-based application with an interactive user interface
2. Register and authenticate users
3. Create, edit, and remove organizations
4. Create, edit, and remove events
5. Manage events and organizations
6. Search for organizations and events
7. Locate events using Google Maps
8. Rate, attend, and host an event
9. Add branches of organizations
10. Endorse other organizations
11. Request representation and/or membership for an organization

# Solution Description

The team implemented a solution which was a web-based application using ASP on the front-end, C# on the back-end, and SQL Server 2012 as the database to store and maintain and store all the information for the application.

## Front-End Description

The team decided to implement ASP as the Graphical User Interface, simplicity of the application being the biggest reason for this. We decided to make the entire web application using drop-down menu and simple textboxes and limiting user input to not allow special characters for security reasons. We also decided to help users by putting default values on the time drop-down and selecting the current date on the calendar. This helped the users save a little time and not give them errors if they don’t select an option. Users also have the option of navigating to any part of the application using the navigation menu and going to the home page where it explicitly lists the functions that can be performed. The users also get a confirmation or the reason of failure for any action performed. They also have the ability to locate events using Google Maps by using the search events feature which is available to non-registered users as well.

## Back-End Description

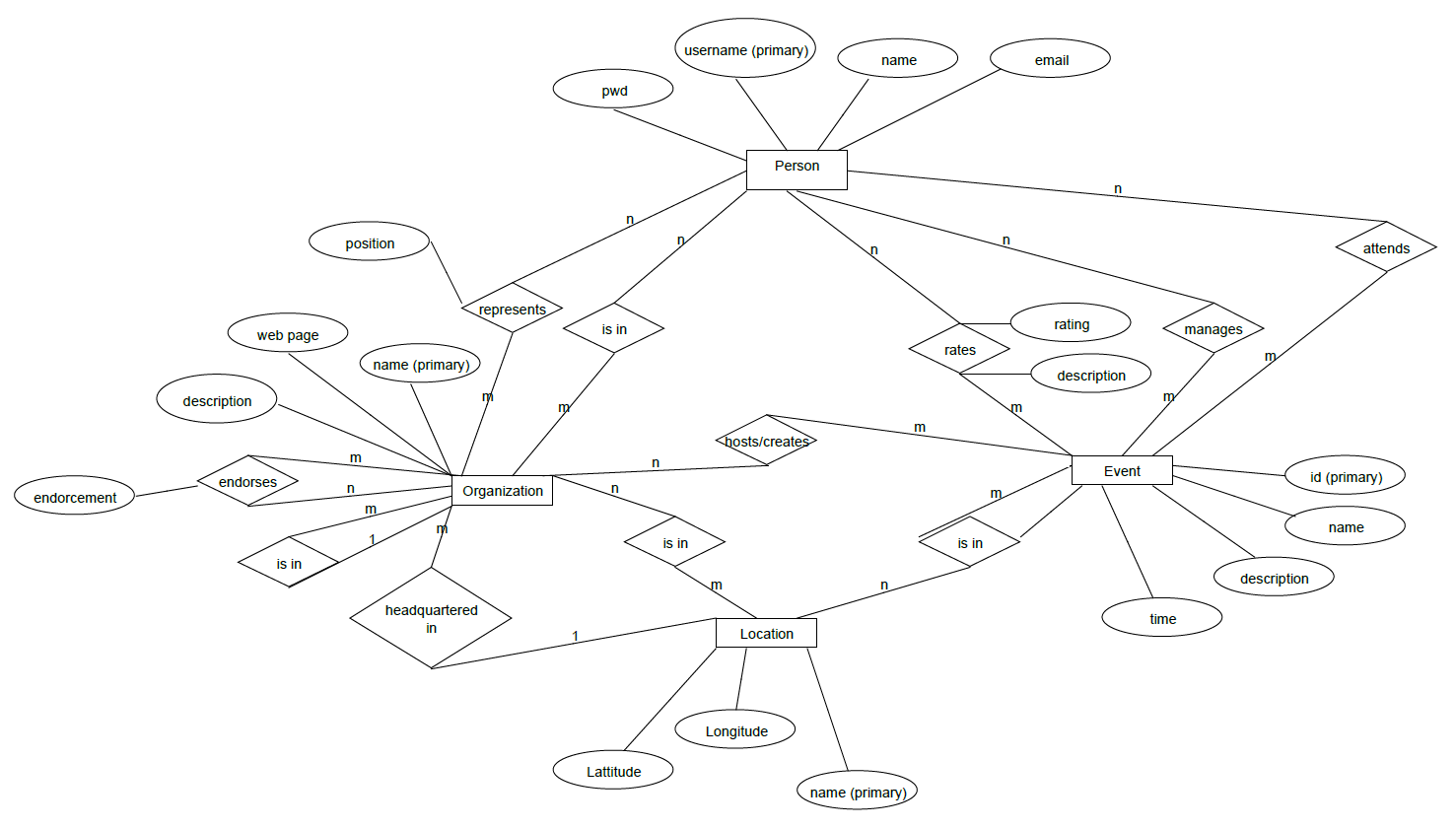
The applications uses the ‘JHIT\_Project43’ database to store all the information and retrieve it. There are numerous tables which contain information about users, organizations, events, locations, and more. Every action performed on the database is done with the help of Stored Procedures. All stored procedures require a username and password validation to perform any changes. The reason for doing this was to make sure only registered users can make changes to the database. Detailed description of how application interacts and provides information from the database can be found in the Database Design section of the document.

# Appendix

## Relational Schema



# Entity-Relationship Diagram



# Explanation of Entity-Relationship Diagram

There are 4 entities in the diagram namely Person, Event, Location, and Organization. Each registered user is supposed to have a username, password and an email address. A user can then log in and create Organizations and/or Events. To create an Event, one must be a representative of an organization. Events and Organizations can have location and headquarter respectively. A user may attend, manage, or rate an event. A person can represent or be a member of an organization. Organizations can also endorse other organizations. Organizations can be in multiple locations, referring to as branches of the organization. Locations have latitude and longitude to locate them on a map.

# Glossary

**Git:** An open source distributed version control system designed to handle projects with speed and efficiency

**Stored Procedure:** Piece of code that performs a specific task each time it is called and is maintained by the database.

**SQL:** A database used to perform actions in databases.

**ASP:** A powerful tool for making dynamic and interactive web pages.

**C#:** Programming language designed for building applications which run .NET framework.

**Entity-Relationship Diagram:** Graphical representation of an information system which shows the relationship between entities like Person, Organization, Events etc.

# Index

Web-Based application: 5

Relational Schema: 3, 4, 7

Entity Relationship diagram: 3,4, 8

Google Maps: 5,6