Travel´s Data Base

For Nomad Republic

Version <1.0>

Bruno Andres de Polignac

Juan Claudio Ollivier Pfister

Table of Contents

1. Introduction 3

1.1 Purpose 3

1.2 Scope 3

1.3 Definitions, Acronyms, and Abbreviations 3

1.4 References 3

1.5 Overview 3

2. Overall Description 3

3. Specific Requirements 3

3.1 Functionality 3

3.1.1 <Functional Requirement One> 3

3.2 Usability 3

3.2.1 <Usability Requirement One> 3

3.3 Reliability 3

3.3.1 <Reliability Requirement One> 3

3.4 Performance 3

3.4.1 <Performance Requirement One> 3

3.5 Supportability 3

3.5.1 <Supportability Requirement One> 3

3.6 Design Constraints 3

3.6.1 <Design Constraint One> 3

3.7 On-line User Documentation and Help System Requirements 3

3.8 Purchased Components 3

3.9 Interfaces 3

3.9.1 User Interfaces 3

3.9.2 Hardware Interfaces 3

3.9.3 Software Interfaces 3

3.9.4 Communications Interfaces 3

3.10 Licensing Requirements 3

3.11 Legal, Copyright, and Other Notices 3

3.12 Applicable Standards 3

4. Supporting Information 3

Software Requirements Specification

# Introduction

Dentro de nuestro proyecto estaremos desarrollando una base de datos para Nomad Republic que es una agencia de viajes la cual cuenta con diferentes paquetes los cuales tienen diferentes destinos.

## Purpose

Desarrollar una base de datos la cual sea capaz de almacenar, modificar y crear nuevas entradas relacionadas con Nomad Republic,.

## Scope

El objetivo es facilitar a el manejo de la información a través de una base de datos optimizando tiempos de respuesta y a su vez tener un mejor control de los movimientos internos de la empresa (Ventas, Ubicaciones, Inventario, etc)

## References

<http://www.nomadrepublic.net>

# Overall Description

La base de datos que estaremos desarrollando será para una empresa que se dedica a organizar viajes alrededor de Mexico. El problema que se desea solucionar con esta base de datos es facilitar la manera en que los organizadores de viajes interactúan con la información de los viajes previamente indexados, de igual forma llevar un control sobre cupos dentro de hoteles, aviones y actividades.

Dentro de la base de datos se tomaran en cuenta varias entidades que tendrán sus propios atributos estas entidades serán:

* Viaje
* Cliente
* Organizador
* Actividades
* Ingresos
* Hoteles
* Grupo
* Transporte
* Empleado

# Specific Requirements

## Dentro de cada viaje un cliente será atendido por un organizador, este organizador le debe de preguntar que actividades desea hacer dentro de su viaje. De igual forma le preguntara que hotel y transporte se acopla a sus necesidades y por ultimo se le asignara un grupo para el viaje que selecciono previamente.

La aplicación debe de ser capaz de capturar dichos procesos entre el organizador y las elecciones del cliente.

## Functionality

Dentro de nuestra base de datos podrá llevar la administración de las entidades previamente mencionadas, de esta manera se llevara un mejor control de los diferentes cambios que pueden existir dentro de ellas, teniendo como restricción los datos de entrada que coincidas con cada tabla para poder en un futuro tener salidas reales de cada tabla.

## Usability

Los usuarios de la base de datos deben de conocer el tipo de datos que maneja cada tabla y sus atributos para eliminar la creación errores dentro de las tablas. De igual forma la manera en que están relacionadas las tablas para poder interactuar entre ellas de una manera mas ágil disminuyendo el numero de clicks que debe de hacer para encontrar la información buscada.

## Reliability

La base de datos debe de estar disponible en todo momento, ya que el organizador o agente en cualquier momento debe de ser capaz de revisar la disponibilidad de cada viaje.

De igual manera los procesos dentro de la base deben de ser precisos para eliminar casos de sobrecupo dentro de algún paquete, actividad, hotel, etc.

Por ultimo debe de existir el índice mas bajo de bugs dentro del código para eliminar errores al momento de entradas, procesamiento o salida.

## Performance

Se tiene planeado optimizar todos los procesos dentro de la base de datos para que la misma trabaje de la manera mas rápida posible, limitando la capacidad de algunas tablas para eliminar tiempos excesivos al momento de trabajar dentro de ella.

## Supportability

Se debe de dar mantenimiento a cada tabla de la base para eliminar sobrantes o extender atributos dentro de ella para poder abarcar todos los aspectos necesarios para la empresa.

## Design Constraints

Para modelar la base de datos utilizaremos DB2, y para hacer la interfaz grafica de la aplicación utilizaremos Ruby on Rails.

## Interfaces

Desarrollaremos una interfaz grafica basada en ruby on rails la cual sea capaz de interactuar con la base de datos, y poder hacer entradas y salidas a la misma.

## Licensing Requirements

Una licencia activa de DB2 de IBM.