|  |
| --- |
| **Vietnam Travel Discovering Project** |
|  |
| **Database Design Document** |
|  |
|  |

**Date: 03-30-2012**

**Version: 2.0**

**Mentor 1: Nguyen Thi Minh Thi**

**Mentor 2: Anh Nguyen Quang**

**VTD TEAMWORK**

Do Thi Thu Hoai

Tran Van Nhan

Che Thi Tu Uyen

Do Tran Viet Cong

Nguyen Thi Xuan Trinh

**VERSION HISTORY**

A-Added M-Modified D-Deleted

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Date | AMD | Changed description | Changed by |
| 1.0 | 03-20-2012 |  | Initiate Document | Che Thi Tu Uyen,  Nguyen Thi Xuan Trinh |
| 2.0 | 03-30-2012 | D    A | - Edit table definition and entity relationship diagram  - Add assumption and constraint, edit entity mapping | Che Thi Tu Uyen |

**Document Approvals:** The following signatures are required for approval of this document.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| *Mentor1:*  *Nguyen Thi Minh Thi* |  | Date: 03/30/2012 |
|  |  |  |
| *Project Manager:*  *Do Thi Thu Hoai* |  | Date: 03/30/2012 |
|  |  |  |
| *Project Members:*  *Nguyen XuanTrinh*  *Che Thi Tu Uyen*  *Do Tran Viet Cong*  *Tran Van Nhan* |  | Date: 03/30/2012 |

Contents

[1.1 Scope, Approach and Methods 5](#_Toc324427070)

[1.2 Related Documents 5](#_Toc324427071)

[2. SUBSYSTEM/APPLICATION OVERVIEW 6](#_Toc324427072)

[2.1 Application overview 6](#_Toc324427073)

[2.2 Hardware and software architecture overview 6](#_Toc324427074)

[2.3 Data store overview 6](#_Toc324427075)

[3. DATABASE DESIGN DECISIONS 6](#_Toc324427076)

[3.1 Assumption and Constraint 6](#_Toc324427077)

[3.2 Entity mapping 7](#_Toc324427078)

[3.2.1. Mapping rules 7](#_Toc324427079)

[3.2.2. Additional Objects 7](#_Toc324427080)

[3.3 Table Definitions 7](#_Toc324427081)

[3.3.1. USER Table 7](#_Toc324427082)

[3.3.2 ROLE Table 8](#_Toc324427083)

[3.3.3 SERVICETYPE Table 8](#_Toc324427084)

[3.3.4 SERVICE Table 8](#_Toc324427085)

[3.3.5. SUBSERVICE Table 8](#_Toc324427086)

[3.3.6 DETAILSERVICE Table 8](#_Toc324427087)

[3.3.7 LOCATION Table 9](#_Toc324427088)

[3.3.8 COMMENT Table 9](#_Toc324427089)

[3.4 Entity Relationship Diagrams 9](#_Toc324427090)

[3.5 Key mappings 12](#_Toc324427091)

[3.6 Dependencies 12](#_Toc324427092)

[3.7 Denormalisation 13](#_Toc324427093)

[3.7.1 Performance Improvement 13](#_Toc324427094)

[3.7.2 Functional Support 13](#_Toc324427095)

[4. IMPLED FUNCTIONALITY 13](#_Toc324427096)

[5. DATA ACCESS 13](#_Toc324427097)

[5.1. Users Privileges 13](#_Toc324427098)

[5.2 Table Access 14](#_Toc324427099)

[6. REFERENCE 15](#_Toc324427100)

1. INTRODUCTION

This Database Design Document provides the basis for the VTD project Database Design. It defines the database that will support the VTD project Data model. It describes both logical and physical definition, non-function issues, and the database interfaces, storage aspects are defined in the physical database design sections. The design is created with expected data volumes, functional and non- functional usage of the tables, and performance considerations and requirements. The following topics are covered in this document:

Assumptions and decisions on database design

* Entity – mapping
* Table, column definitions
* Primary, unique and foreign key definitions
* Column and row level validation rules (check constraints)
* Rules for populating specific columns (sequences, derivations, demoralized columns)
* Interfaces and dependencies with other components
* Data access description.

## Scope, Approach and Methods

The Database Design for the VTD project is composed of definitions for database object derived by mapping entities to tables attributes columns, unique identifies to unique keys and relationship to foreign keys.

During design, these initial definitions are enhanced to support the functionality described in the functional specification / user stories and defined in the primary and supporting modules of the application high level design.

## Related Documents

This specification refers to the following documents:

* + - VTD Proposal
    - VTD Plan
    - VTD Architecture Document
    - VTD Detail Design Document

# SUBSYSTEM/APPLICATION OVERVIEW

## Application overview

This website provides information for those who want to travel or find out information on hotels, restaurants and services for shopping, beauty and entertainment

The traveler can post information to share for someone who visited the website, and can give comment to information about services.

Administrator can manage information about services and manage user information

## Hardware and software architecture overview

This section provides an overview of the software and hardware architecture. The following describes the technology components of the VTD project.

The technology components of Website for managing database

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| Design | Website |
| Platform | .Net framework |
| Database | Database Management System MS SQL Server |
| Software | C#, ASP.Net, Visual studio 2008 |
| Hardware | PC with Windows 2000 or more, Intel core 2 duo CPU, 2GB RAM |

## 

## Data store overview

This session briefly introduces all data stores including database, file systems.

The VTD project stores data in SQL Server database

# DATABASE DESIGN DECISIONS

This section contains the decisions that were made when designing the database for the VTD project. Problem, alternative solutions and motivated choices are listed below. The section also lists any design assumptions that had to be made. In case the assumptions are results of ambiguities or lack of details, they will need verifying by the analyst team.

## Assumption and Constraint

The following constraints and assumptions will be imposed on the project during the whole duration of the project.

Assumption

* + This project follows modified waterfall model

Constraint

* + Travelling information is uploaded by all Members who logged in system
  + Travelling information is publicized after approved by Mod or Admin
  + All Members can comment for travelling services
  + User name’s minimum size is 8 characters, without special character
  + Password’ minimum size is 4 characters. Password must contain both letters and digits

## Entity mapping

### 3.2.1. Mapping rules

When mapping entities to tables, the following rules were applied:

* Entities are mapped into tables in a one to one manner.
* Attributes are mapped to columns in a one to one manner.
* One-to-many relationships are mapped to foreign keys.
* Many-to-many relationship are implemented using two one-to-many relationships with intersection table

### 3.2.2. Additional Objects

The following table lists database objects (tables or columns) that don’t derived from an entity, but were added to the database design for the purpose listed below. This includes intersection tables used for mapping many-to-many relationships.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Table** | **Mapped from** | **Purpose** |
|  | COMMENT | Many-to-many relationship of USER table and DETAILSERVICE Table | Specify a user can have many comments for a detail service, and a detail service is commented by many users |

## 

## Table Definitions

### 3.3.1. USER Table

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Description** | **Data type** | **Size** |
| **Username** | User index | Nvachar | 50 |
| Password | Password of user | Nvachar | 50 |
| Email | Email of user | Nvachar | 50 |
| RegistrationDate | RegistrationDate of user | Date |  |
| Disable | Disable | Bit |  |
| RoleID | Position index | Nvarchar | 10 |

### 

### 3.3.2 ROLE Table

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Description** | **Data type** | **Size** |
| **RoleID** | Position index | Nvachar | 10 |
| RoleName | Name of position | Nvachar | 50 |

### 

### 3.3.3 SERVICETYPE Table

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Description** | **Data type** | **Size** |
| **ServiceTypeID** | Service index | Nvachar | 10 |
| ServiceTypeName | ServiceType\_Name | Nvachar | 50 |
| Image | Image of service type | Text |  |

### 3.3.4 SERVICE Table

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Description** | **Data type** | **Size** |
| **ServiceID** | Service index | Nvachar | 10 |
| ServiceName | Name of Service | Nvachar | 50 |
| ***ServiceTypeID*** | ServiceType index | Nvachar | 10 |
| Image | Image of service | Text |  |
|  |  |  |  |

### 3.3.5. SUBSERVICE Table

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Description** | **Data type** | **Size** |
| **SubServiceID** | SubService index | Nvachar | 10 |
| SubserviceName | Name of SubserviceName | Nvachar | 50 |
| ***ServiceID*** | Service index | Nvachar | 10 |

### 

### 3.3.6 DETAILSERVICE Table

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Description** | **Data type** | **Size** |
| **DetailServiceID** | Detail service index | Nvachar | 10 |
| DetaiServiceName | Name of detail service | Nvachar | 50 |
| Address | Address of service | Integer | 50 |
| Description | Description Service | Text |  |
| Image | Image Of Service | Text |  |
| PhoneNumber | Phone number of service | Nvachar | 20 |
| ***SubserviceID*** | Subservice index | Nvachar | 10 |
| ***LocationID*** | Location index | Nvachar | 10 |
| ***ServiceID*** | service index | Nvachar | 10 |
| DetailServiceDate | Date of Detail Service | Datetime |  |
|  |  |  |  |

### 3.3.7 LOCATION Table

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Description** | **Data type** | **Size** |
| **LocationID** | Location index | Nvachar | 10 |
| LocationName | Name of location | Nvachar | 50 |

### 

### 3.3.8 COMMENT Table

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Description** | **Data type** | **Size** |
| **CommentID** | Comment index | Nvachar | 10 |
| CommentName | Name of comment | Nvachar | 50 |
| Content | Content comment of user | Text |  |
| DateComment | Date comment | Date |  |
| Uername | Name of user comment | Nvachar | 50 |
| ***DetailServiceID*** | DetailService ID | Nvachar | 10 |

**Comment:**

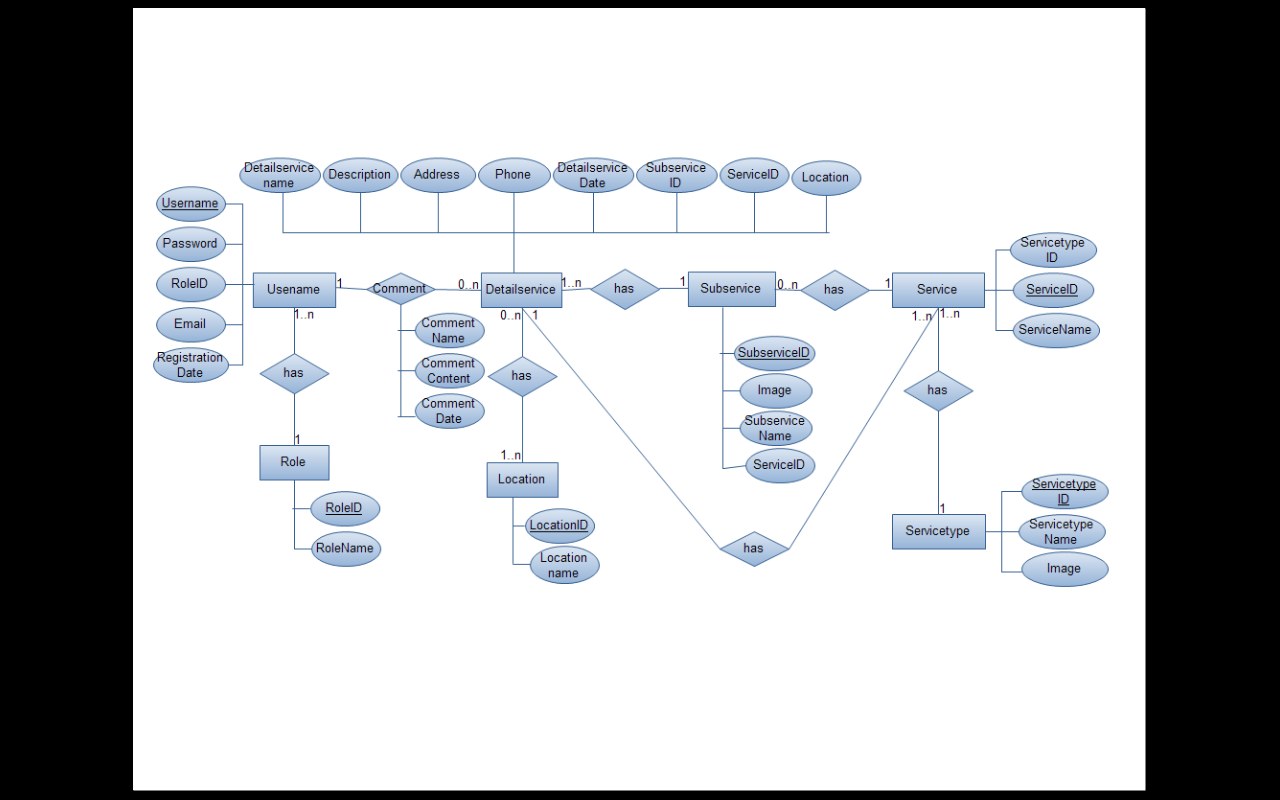
Primary key is printed in underline, bold

Foreign key is printed in italics, bold

## 3.4 Entity Relationship Diagrams

The following diagram shows the entities of the VTD System schema and their relationships

**3.4.1 Associate** **Entity- model**



**3.4.2** **Entity-relationship model**

## C:\Users\Che Uyen\Desktop\PhotoShare.png

## 3.5 Key mappings

The following tables have primary keys created from sequences:

|  |  |  |
| --- | --- | --- |
| **Table** | **Primary key column** | **Sequence** |
| ROLE | RoleID | NVarchar |
| USER | Username | Nvarchar |
| SERVICE | ServiceID | Nvarchar |
| SUBSERVICE | SubServiceID | Nvarchar |
| DETAILSERVICE | DetailServiceID | Nvarchar |
| LOCATION | LocationID | Nvarchar |
| SERVICETYPE | ServiceTypeID | Nvarchar |
| COMMENT | CommentID | Nvarchar |

## 

## 3.6 Dependencies

List here any dependencies for the VTD System schema. One type of dependencies can be foreign keys across schemas. One-to-many relationships are mapped to foreign keys. List foreign key dependencies here:

|  |  |  |
| --- | --- | --- |
| **Table/column** | **Table/column refers to** | **Comment** |
| ROLE/  RoleID | USER/  RoleID |  |
| COMMENT/  CommentID | USER/  CommentID |  |
| SERVICETYPE/  ServiceTypeID | SERVICE/  ServiceTypeID |  |
| SERVICE/  ServiceID | SUBSERVICE/  ServiceID |  |
| SERVICE/  ServiceID | DETAILSERVICE/  ServiceID |  |
| LOCATION/  LocationID | SERVICE/  LocationID |  |
| LOCATION/  LocationID | SUBSERVICE/  LocationID |  |
| SUBSERVICE/  SubserviceID | DETAILSERVICE/  SubserviceID |  |

## 3.7 Denormalisation

To improve performance or otherwise support specific functionality, redundancy is sometimes added to the design. Two types of redundancy are distinguished, performance denormalisation and functional denormalisation. The first type is aimed at improving performance; the second is needed to support the proposed functionality of the system.

### 3.7.1 Performance Improvement

To maintain redundant or denormalisation data needed to improve performance, the following objects were modified:

|  |  |  |
| --- | --- | --- |
| **Denotation** | **Source table or entity** | **Rules and methods for maintaining integrity** |
|  |  |  |

### 3.7.2 Functional Support

To maintain redundant or denormalised data needed to support the proposed functionality of the following objects were modified:

|  |  |  |
| --- | --- | --- |
| **Demoralized Table/Column** | **Source table or entity** | **Rules and methods for maintaining integrity** |
|  |  |  |

# IMPLED FUNCTIONALITY

The model contains supporting modules that perform functions that may be trivial to the users but are considered to be important functionality by the developers

# DATA ACCESS

## 5.1. Users Privileges

The following users are recognized as being required.

Users Privileges are sorted from low to high:

|  |  |
| --- | --- |
| **User name** | **Purpose** |
| Guest | Search, view information |
| Member | Upload and comment detail services |
| Moderator | Manage service information and service type |
| Admin | Manage user information and authorize for users |

## 

## 5.2 Table Access

Below is a list of particularly performance-critical functions and their table usage

|  |  |  |  |
| --- | --- | --- | --- |
| **Function** | **Peak Frequency** | **Tables used** | **Access user** |
| Register | Add, view | USER | Guest |
| Login | Add | USER | Member, Moderator, Admin |
| Logout | Read only | USER | Member, Moderator, Admin |
| Search information | Read only | USER, SERVICE | Any user |
| View information | Read only | USER, SERVICE, SUBSERVICE, DETAILSERVICE | Any user |
| Comment | Add, delete, edit | COMMENT | Admin, Moderator,  Member |
| Manager user information | Add, delete, edit | USER | Admin |
| Update user information | Add | USER | Admin, Moderator, Member |
| Upload service information | Add, delete, edit | DETAILSERVICE | Admin, Moderator, Member |
| Approve service information | Add, delete | DETAILSERVICE | Admin, Moderator |
| Manage service type | Add, delete, edit | SERVICETYPE, SERVICE, SUBSERVICE | Mod, Admin |
| Authorize | Add, delete | USER | Admin |

# REFERENCE

* VTD Proposal
* VTD Project Plan
* VTD Architecture Document
* VTD Detail Design Document