

**International School**

**Capstone Project 2**

**CMU-SE 451 – C2SE.12**

**Database Design Document**

**Version 1.0**

**Date: March 20th, 2021**

**Learning English Together**

**Submitted by**

**Ha, Le Thanh**

**Hieu, Le Xuan**

**My, Ngo Ngoc**

**Thong, Doan Trung**

**Approved by**

**MSc Huy, Truong Dinh**

**Proposal Review Panel Representative:**

Name Signature Date

**Capstone Project 2- Mentor:**

Name Signature Date

**PROJECT INFORMATION**

|  |  |  |  |
| --- | --- | --- | --- |
| **Project acronym** | E2 | | |
| **Project Title** | Easy English | | |
| **Start Date** | 26 Feb 2021 | **End Date** | 08 Jun 2021 |
| **Lead Institution** | International School, Duy Tan University | | |
| **Project Mentor** | MSc Huy, Truong Dinh | | |
| **Scrum master / Project Leader & contact details** | Ha, Le Thanh  Email: lethanhhadtu@gmail.com  Tel: 0334002818 | | |
| **Partner Organization** | Duy Tan University | | |
| **Project Web URL** |  | | |
| **Team members** | Name | Email | Tel |
|  | Ha, Le Thanh | lethanhhadtu@gmail.com | 0334002818 |
|  | Hieu, Le Xuan | xuanhieu.le.1999@gmail.com | 0399706614 |
|  | My, Ngo Ngoc | ngongocmy851999@gmail.com | 0764497391 |
|  | Thong, Doan Trung | doanthong002@gmail.com | 0886428208 |

REVISION HISTORY

| **Version** | **Date** | **Comments** | **Author** | **Approval** |
| --- | --- | --- | --- | --- |
| 1.0 | March 6th, 2021 | Initial Release | H3T Team |  |

TABLE OF CONTENT

[1. Introduction 5](#_Toc66220016)

[**1.1. Purpose Of Document 5**](#_Toc66220017)

[**1.2. Project Goal 5**](#_Toc66220018)

[2. Problem Definition 5](#_Toc66220019)

[**2.1. Business need 5**](#_Toc66220020)

[**2.2. Solution 6**](#_Toc66220021)

[3. Current Status of Art 6](#_Toc66220022)

[4. Engineering Approach 7](#_Toc66220023)

[**4.2. System context description 7**](#_Toc66220024)

[**4.3. Technical Constraints 8**](#_Toc66220025)

[5. Tasks and Deliverables 10](#_Toc66220026)

[6. Project Management 11](#_Toc66220027)

[**6.1. Cost/Budget for Project 11**](#_Toc66220028)

[**6.2. Tentative Schedule 12**](#_Toc66220029)

[**6.2.1. Master Plan 12**](#_Toc66220030)

[**6.2.3. Scrum Process 13**](#_Toc66220031)

[7. Project Constraints 14](#_Toc66220032)

[8. Conclusion 16](#_Toc66220033)

[9. References 16](#_Toc66220034)

[10. Attachment 16](#_Toc66220035)

1. Introduction

The Database Design maps the logical data model to the target database management system with consideration to the system’s performance requirements. The Database Design converts logical or conceptual data constructs to physical data constructs (e.g., tables...) of the target Database Management System.

* 1. Purpose

The purpose of the Database Design is to ensure that every database transaction meets or exceeds its performance requirements. This document takes into account data and transaction volume to produce a schema and environment that will meet necessary

performance

* 1. Scope

The Database Design Document has the following objectives:

● To describe the design of a database, that is, a collection of related data stored in one or more computerized files that can be accessed by users or developers via a DBMS

● To serve as a basis for implementing the database and related software units. It provides the acquirer visibility into the design and provides information necessary for software development.

**1.3 Introduction about MongoDB**

MongoDB is a NoSQL database which stores the data in form of key-value database and can be installed across different platforms like Windows, Linux etc. pairs. It is an **Open Source**, **Document Database** which provides high performance and scalability along with data modelling and data management of huge sets of data in an enterprise application.

MongoDB also provides the feature of Auto-Scaling. Since, MongoDB is a cross platform like Windows, Linux etc.

A Document is nothing but a data structure with name-value pairs like in JSON. It is very easy to map any custom Object of any programming language with a MongoDB Document. For example:**Student** object has attributes **name,** **rollno** and **subjects**, where subjects is a List.

Document for Student in MongoDB will be like:

{

name : "Stduytonight",

rollno : 1,

subjects : ["C Language", "C++", "Core Java"]

}

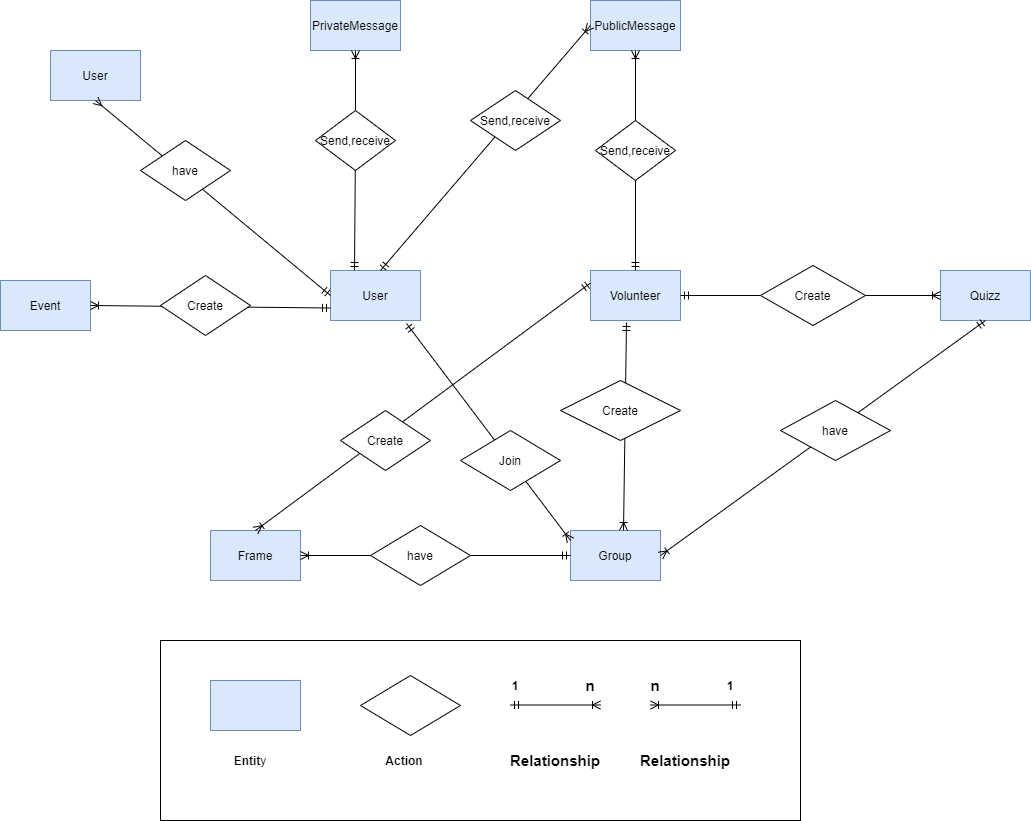
We can see, Documents are actually JSON representation of custom Objects. Also, excessive JOINS can be avoided by saving data in form of Arrays and Documents(Embedded) inside a Document.

**2.Database Diagram**

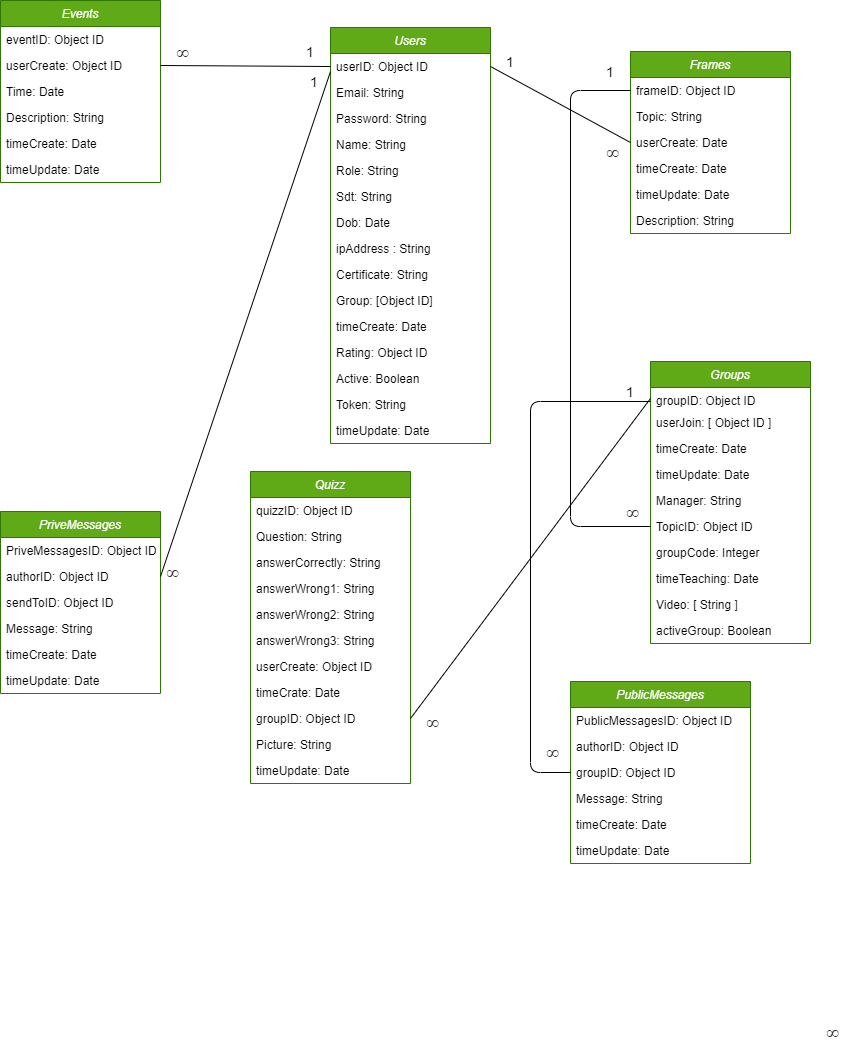
**2.1 Table Overview**

|  |  |
| --- | --- |
| **Table Name** | **Short Description** |
| Group | This table shows group information including all related fields. |
| User | This table shows member information including all related fields. |
| Frame | This table shows topic of a group |
| PrivateMessage | This table shows messages between members. |
| PublicMessage | This table shows messages between User in the group. |
| Quizz | This table shows quiz of the group |
| Event | This table shows all the events for the website. |

**2.2 Entity Relationship Diagram**

****

**2.3 Table Relationship Diagram**

****

3. Database Design for Sprint

3.1 Table Group

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Group | | | | |
| Field | Type | Constrain | Nullable | Description |
| \_id | Object | Primary Key |  | the MongoDB driver automatically generates an ObjectId |
| managerId | Object |  |  | Id Manager Group |
| topicId | Object |  |  | Id Topic Group |
| groupCode | String |  |  | Code Group |
| timeTeaching | date |  |  | Time Teach |
| videoLink | String |  |  | Link Video Group |
| action | Boolean |  |  | Action of group |
| timeCreate | Date |  |  | Time Create |
| timeUpdate | Date |  |  | Time Update |
| userCreate | Object |  |  | Id User Create Group |

3.2 Table User

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| User | | | | |
| Field | Type | Constrain | Nullable | Description |
| \_id | Object | Primary key |  | Id Mongodb Create |
| Name | String |  |  | Name of user . Display on website. |
| password | String |  |  | Password of user. password after saving to db will be encrypted |
| email | String |  |  | Connect with user and help user get their password back. Its value must be unique in the collection |
| sex | String |  |  | Sex of user. |
| dob | Date |  |  | Date of birth of user |
| role | String |  |  | user authorization. Three type: 1. Admin 2. Volunteer 3. Student |
| action | Boolean |  |  | Status of user |
| numberPhone | String |  |  | Phone of user |
| token | String |  |  | Token of user device |
| avatar | String |  |  | Avatar of user. Save the path to cloud nary. Display on website. |
| ratings | Object |  |  | Rank of user |
| timeCreate | Date |  |  | User Time create |
| timeUpdate | Date |  |  | User Time Update |
| address | String |  |  | Address of user |
| group | Object |  |  | All group user actives |
| certificate | String |  |  | Certificate of user |

3.3 Table Frame

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Frame | | | | |
| Field | Type | Constrain | Nullable | Description |
| \_id | Object | Primary key |  | Id Mongodb Create |
| topic | String |  |  | Name Topic |
| userCreate | Object |  |  | Id User Create |
| description | String |  |  | Content Topic |
| timeCreate | Number |  |  | Time Create |
| timeUpdate | Date |  |  | Time Update |
| action | Boolean |  |  | Status of Topic |

3.4 Table PrivateMessage

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PrivateMessage | | | | |
| Field | Type | Constrain | Nullable | Description |
| \_id | Object | Primary key |  | Id Mongodb Create |
| authorId | Object |  |  | Id Sender |
| sendToId | Object |  |  | Id Receiver |
| message | String |  |  | Content Message |
| timeCreate | Number |  |  | Time Create |
| timeSend | Date |  |  | Time when message is sent |
| uniqueId | Number |  |  | Private Message ID |

3.5 Table PublicMessage

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PublicMessage | | | | |
| Field | Type | Constrain | Nullable | Description |
| \_id | Object | Primary Key |  | the MongoDB driver automatically generates an ObjectId. It is ID of message. |
| authorId | Object |  |  | Id of Sender |
| group | Object |  |  | Group Name of public message |
| message | String |  |  | Content of message |
| timeCreate | Number |  |  | Time Create |
| timeSend | Date |  |  | Time when message is sent |

3.6 Table Events

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Event | | | | |
| Field | Type | Constrain | Nullable | Description |
| \_id | Object | Primary key |  | the MongoDB driver automatically generates an ObjectId. It is ID of event. |
| userCreate | Object |  |  | Event ID |
| timeEvent | Date |  |  | Event Time |
| description | String |  |  | Event Content |
| timeCreate | Nummber |  |  | Event Time Create |
| timeUpdate | Date |  |  | Event Time Update |

3.7 Table Quizz

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Quizz | | | | |
| Field | Type | Constrain | Nullable | Description |
| \_id | Object | Primary key |  | the MongoDB driver automatically generates an ObjectId. It is ID of quizz. Each quizz will create 1 quiz by id |
| question | String |  |  | Content Question |
| answerCorrectly | String |  |  | Answer correct |
| answerWrong1 | String |  |  | Answer wrong 1 |
| answerWrong2 | String |  |  | Answer wrong 2 |
| answerWrong3 | String |  |  | Answer wrong 3 |
| userCreate | Object |  |  | Quiz Id User Create |
| Picture | String |  |  | Quiz Picture |
| timeCreate | Number |  |  | Quiz Time Create |
| timeUpdate | Date |  |  | Quiz Time Update |
| groupId | Object |  |  | Quiz Id Group |

4.Hardware and software Requirements

This section provides an overview of hardware and software requirements. Below are descriptions of the technological components of the Easy English Website:

|  |  |
| --- | --- |
| **Attributes of Easy English WEBSITE** | |
| **Attributes** | **Descriptions** |
| **Database** | MongoDB |
| **Software** | Reactjs, Nodejs, Polling, Elasticsearch, Redis, Websocket, Mail gun |
| **Hardware** | Computer |
| **Library** | Hook, Redux, React hook form, Bootstrap 4, Fontawesome, Material ui, Express.js. |