Progress Report

TutorHere

Name : Sathischandra M.K.D.T.M.

Registration No. : EN82722

Index No. : 16/ENG/093

Course Code : CO3302

Field of Specialization: Computer Engineering

Introduction

"TutorHere" is a tutor hiring online service which provides facilities for the students or the learners to find the tutors for a required subjects of study and hire them for tutoring. In the same time, this platform provides the ability for the tutors to connect to the tutor seekers with less effort and find tutoring opportunities. TutorHere provides ranked tutors based on set of criteria for the user request and the user will be able to hire the tutor. Based on the hourly rate defined by the tutor and the time of work, the fee is calculated and payment can be done.

Completed Work of the Project

The 'Analysis' and the 'Design' phases of the project are completed and currently, the 'Implementation' phase is being worked on. The requirement gathering and the requirement analysis parts are completed successfully. After that, the database design was completed. Entity relational model of the project is shown in Figure 1.1.

Design of the Application.

The application contains two main entities named "Tutor" and "Learner". The learner can request a tutor for the subject that he/she wish to learn. The tutor can define his/her affordable time periods and there are two entities named "SingleTimeSlot" and "WeeklyTimeSlot" for setting tutor's time-slots. Single time-slot is a one-time time period and in contrast, weekly time-slot is a weekly repeating time period.

E.g.: Single time-slot: 4th September 2019, 08:00 am to 02:00 pm

Weekly time-slot: Wednesday, 07:00 am to 11:00 am

Learner is the one who can request for tuitions. First, the learner needs to enter subject, time period and location. Then, the application will show a list of most relevant tutors for the selected criteria. "TutorLocations" and "LearnerLocation" entities store the location data needs for finding the tutors within minimum distance. "Subject" entity holds tutors' subject information.

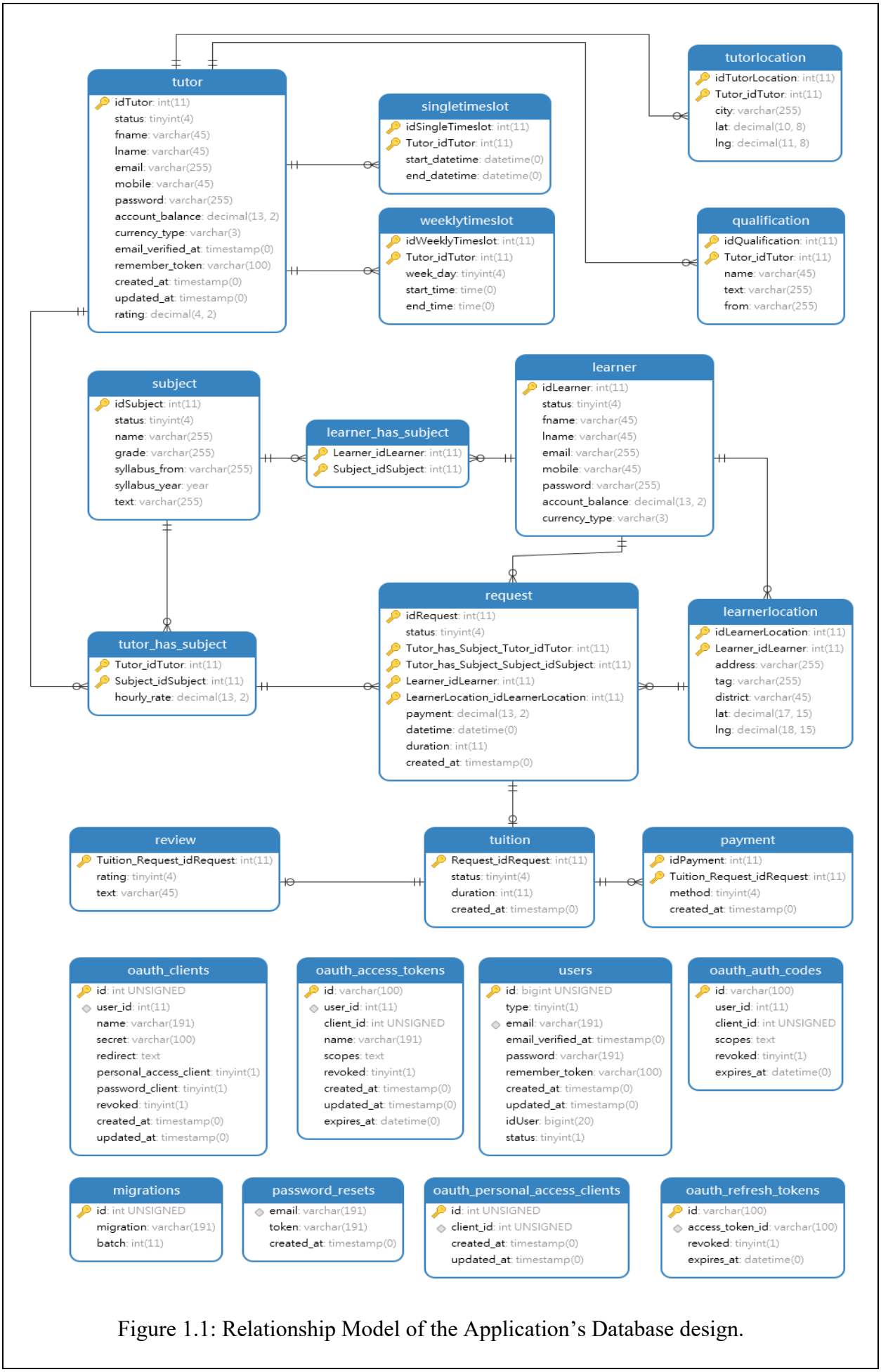

Figure 1.1: Relationship Model of the Application's Database design.

After learner, select the tutor, a new “request” for a tuition is created. Then, if the tutor accept the request, new “tuition” is created. Then the tuition will be conducted. “Payment” and “Review” are the entities that holds payments information and learner reviews of the tuition, respectively.

Implementation on the front-end and the back-end is currently going on parallely. Back-end is implemented in PHP language (using Laravel framework) and front-end is implemented using Javascript framework named VueJS. Backend provides RESTful API methods to communicate between the front-end and the back-end.

Front-end is being developed component by component and after finishing the development of every single component, the final interface will be created by combining each component. When each component was created, the API methods and back-end functionalities of those component were also created simultaneously.

This applications is consisted with 2 main views namely ‘TutorView’ and ‘LearnerView’. The authentication processes of both views are almost completed and the user registration components and log-in components have been created. Then, basic components for updating user settings of both Tutor and Learner are created. In this application, the Learner and Tutor locations play an important role in filtering the appropriate tutors for the learners requests. Therefore, next, the integration of Google Map API and development of other location related functionalities were carried on. Next, the components for finding, filtering appropriate tutors and requesting a tuition were created.

Work to be completed of the Project

The tutor’s view’s components such as showing the requested tuitions, showing learners informations have not been created yet. And the other challenging part is, real-time updating of the tutor’s view according to the user requests.

In this version of the application, the payment is handled only by cash (Online payments will be added in future versions). Therefore, creating the payment handling component will not be harder. And the tuition review submission component also has to be completed.

In this application, when learner creating a tuition request, the tutor listing and ranking system is bit more challenging. Implementation of some machine learning algorithms for this, will also be considered. And finally, all those individually created components have to be merged together to form the final interfaces of the application.