 **MINISTRY OF EDUCATION AND TRAINING**

**FPT UNIVERSITY**

Capstone Project Document

**Mobile Based Language Learning Application**

|  |  |
| --- | --- |
| **Group 1** | |
| **Group members** | Lê Ngọc Hiếu – Team Leader – SE61723  Đỗ Hoàng Nguyên – Team Member – SE61556  Nguyễn Nhật Quang – Team Member – SE61472  Trương Thanh Lâm – Team Member – SE61257 |
| **Supervisor** | Mr. Nguyễn Huy Hùng |
| **Ext. Supervisor** | N/A |
| **Capstone Project code** | MOLA |

- Ho Chi Minh City, 05/2017 -

*This page is intentionally left blank*

**

**CAPSTONE PROJECT REGISTER**

Class: SE0970 Duration time: From 08/05/2017 To 14/8/2017

(\*) Profession: <Software Engineer> Specialty: <ES> <IS>

x

(\*) Kinds of person make registers: Lecturer Students

x

1. Register information for supervisor (if have)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Full name** | **Phone** | **E-Mail** | **Title** |
| Supervisor 1 | Nguyễn Huy Hùng |  | hungnh@fpt.edu.vn | Mr. |

2. Register information for students (if have)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Full name** | **Student code** | **Phone** | **E-mail** | **Role in Group** |
| Student 1 | Lê Ngọc Hiếu | SE61723 | 0907769634 | Hieulnse61723@fpt.edu.vn | Leader |
| Student 2 | Nguyễn Nhật Quang | SE61472 | 0974345404 | Quangnnse61472@fpt.edu.vn | Member |
| Student 3 | Đỗ Hoàng Nguyên | SE61556 | 0914834032 | Nguyendhse61556@fpt.edu.vn | Member |
| Student 4 | Trương Thanh Lâm | SE61257 | 01663253023 | Lamttse61257@fpt.edu.vn | Member |

3. Register content of Capstone Project

(\*) 3.1. Capstone Project name:

English: Mobile Based Language Learning Application

Vietnamese: Ứng Dụng Học Ngôn Ngữ Trên Thiết Bị Di Động

Abbreviation: MOLA

(\*) 3.2. Main proposal content (including result and product)

1. Theory and practice (document):
   * Student should apply the software development process and the UML 2.0 in modeling the system
   * Software artifacts include User Requirement, Software Requirement Specification, Architecture Design, Detail Design, System Implementation and Testing Document, Installation Guide, sources code, and deployable software packages
   * Server side technique:
     + Database design, OOA, OOD, OOP, MVC, Java or .Net technology, …
     + Apache Lucene, Elastic Search
   * Client side technique
     + HTML5, CSS, JavaScript, JQuery, Ajax
     + Mobile Platform (iOS, Android)
2. Program:

Building an application that allows users to learning and teach languages via mobile devices. The main purposes of this application are:

* allow users to practice his/her speaking skill with native speakers via mobile devices.
* connect people who want to learn and who want to teach

The following main features should be implemented:

* Register time slots: users who want to teach can register his/her available time slots so that other users can register to learn
* Course Tool: a simple tool should be supported so that teachers can compose a course structure
* Effective score: each teacher can have effective score which indicates how effective he/she has been teaching. The effective score is calculated based on many factors: punctuality (i.e online at the registered time slots), rating from learners….
* Supported learning tools: while practicing speaking with teachers, learners can be supported with tools such as dictionary, pronunciation supports…
* Learner Ranking: each learner will have a ranking score which indicates the progress he/she has made
* Privacy control: users should be able to control his/her camera viewing from other
* Keep tracking learning progress
* Suggestion: learners should be able to view suggested teachers and learners

1. Other products:

* All of management functions of the system must be implemented to support the operating system.

4. Other comment (propose all relative thing if have)

N/A

|  |  |
| --- | --- |
| **Supervisor (If have)**  *(Sign and full name)* | HCM city, date 04/05/2017 …..  **On behalf of Registers**  *(Sign and full name)* |

# Table of Contents

[A. Introduction 1](#_Toc490427359)

[1. Project Overview 1](#_Toc490427360)

[2. Introduction 1](#_Toc490427361)

[3. Current Situation 1](#_Toc490427362)

[4. Problem Definition 1](#_Toc490427363)

[5. Proposed Solution 2](#_Toc490427364)

[5.1 Featured functions 2](#_Toc490427365)

[5.2 Advantages 2](#_Toc490427366)

[5.3 Disadvantages 2](#_Toc490427370)

[B. Software Project Management Plan 2](#_Toc490427373)

[1. Problem Definition 2](#_Toc490427374)

[1.1 Name of this Capstone Project 2](#_Toc490427375)

[1.2 Problem Abstract 2](#_Toc490427376)

[1.3 Project Overview 3](#_Toc490427377)

[2. Project organization 5](#_Toc490427378)

[2.1 Software Process Model 5](#_Toc490427379)

[2.2 Roles and Responsibilities 6](#_Toc490427380)

[2.3 Tools and Techniques 6](#_Toc490427381)

[3. Project Management Plan 7](#_Toc490427382)

[3.1 Product Backlog 7](#_Toc490427383)

[3.2 Sprint Backlog 7](#_Toc490427384)

[3.3 Deliverables 7](#_Toc490427385)

[3.4 All Meeting Minutes 7](#_Toc490427386)

[C. Software Requirement Specifications 8](#_Toc490427387)

[1. User Requirement Specification 8](#_Toc490427388)

[1.1 Guest Requirement 8](#_Toc490427389)

[1.2 Learner Requirement 8](#_Toc490427390)

[1.3 Teacher Requirement 8](#_Toc490427391)

[1.4 Staff Requirement 8](#_Toc490427392)

[1.5 Admin Requirement 8](#_Toc490427393)

[2. System Requirement Specification 9](#_Toc490427394)

[2.1 System Overview Use Case 9](#_Toc490427395)

[3. Conceptual Diagram 10](#_Toc490427396)

[D. Software Design Description 12](#_Toc490427397)

[1. Design Overview 12](#_Toc490427398)

[2. System Architectural Design 12](#_Toc490427399)

[2.1 Web service architecture description 12](#_Toc490427400)

[2.2 Mobile Application Architecture Description 13](#_Toc490427401)

[3. Component Diagram 14](#_Toc490427402)

[4. Detailed Description 17](#_Toc490427403)

[4.1 Class Diagram 17](#_Toc490427404)

[5. Database Design 19](#_Toc490427405)

[5.1 Entity Relationship Diagram (ERD) 19](#_Toc490427406)

[5.2 Data Dictionary 20](#_Toc490427407)

[6. Algorithms 20](#_Toc490427408)

[6.1 Recommendation 20](#_Toc490427409)

[E. Tasksheet 28](#_Toc490427410)

[F. Appendix 32](#_Toc490427411)

[1. UML Documentation of IBM 32](#_Toc490427412)

[2. React Native 32](#_Toc490427413)

[3. WebRTC 32](#_Toc490427414)

[4. Play Framework 2.5 32](#_Toc490427415)

[5. Redis Cache 32](#_Toc490427416)

[6. Firebase Cloud Message 32](#_Toc490427417)

[7. Firebase Realtime Database 32](#_Toc490427418)

[8. Dictionary API 32](#_Toc490427419)

# List of Tables

[*Table 1: Definitions, Acronyms, and Abbreviations* 11](#_Toc490226345)

[*Table 2: Roles and Responsibilities* 17](#_Toc490226346)

[*Table 3: Tools and Techniques* 18](#_Toc490226347)

[*Table 4: Deliverables* 18](#_Toc490226348)

[*Table 5: Conceptual Diagram Data Dictionary* 22](#_Toc490226349)

[*Table 6: Component Diagram Dictionary* 27](#_Toc490226350)

[*Table 7: Class Diagram Dictionary* 29](#_Toc490226351)

[*Table 8: ERD dictionary* 31](#_Toc490226352)

# List of Figures

[*Figure 1: Scrum Process* 16](#_Toc490226092)

[*Figure 2: Use case overview* 20](#_Toc490226093)

[*Figure 3: Conceptual Diagram* 21](#_Toc490226094)

[*Figure 4: Web service Architecture* 24](#_Toc490226095)

[*Figure 5: Mobile Application Architecture* 25](#_Toc490226096)

[*Figure 6: Component Diagram* 26](#_Toc490226097)

[*Figure 7: Class Diagram* 28](#_Toc490226098)

[*Figure 8: Entity Relationship Diagram* 30](#_Toc490226099)

[*Figure 9: Percentage of completed lessons flow chart* 36](#_Toc490226100)

[*Figure 10: Build Dataset Flow Chart* 36](#_Toc490226101)

[*Figure 11: Build New User Dataset Flow Chart* 37](#_Toc490226102)

[*Figure 12: User- based Recommendation Flow Chart* 38](#_Toc490226103)

# Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| MOLA | Mobile Based Language Learning Application |
| UI | User Interface |
| UX | User Experience |
| MVVM | Model – View – View Model |
| ORM | Object Relation Mapping |
| TDD | Test Driven Development |

*Table 1: Definitions, Acronyms, and Abbreviation*

# Introduction

## Project Overview

* Project name: Mobile Based Language Learning Application
* Project code: MOLA
* Product type: mobile application
* Start date: 10/05/2017
* End date:

## Introduction

In this document, we introduce a solution for learning and teaching languages via mobile devices. Because of globalization, using different languages for communication becomes essential. The demand for learning languages therefore grows day-by-day. With the help of information technology, many supporting tools for learning languages, such as dictionaries or remote/self learning websites, appear.

The system helps connect people who want to teach or learn languages. We will concentrate on supporting speaking activities between teachers and learners. Practicing speaking skills with native speakers may provide learners with great improvement. In reverse, those “native speakers” teachers may be learners in other languages as well.

This document also describes our working process in 4 months includes our perspective in the system, component designs and detailed core workflows. We hope the system and our solution will help people who are into languages learn effectively.

## Current Situation

People may learn foreign languages at school, university, in a language center or even via the Internet. In those methods, learners may not always interact with native speakers. Moreover, those methods are not flexible in some aspects such as time, content or the language itself.

## Problem Definition

Below are disadvantages of current situation:

* Learners get balanced practicing. However, their ability may be different among skills. There are people who are good in reading or writing but not good in listening and speaking, and they want to improve those latter skills.
* Learners may be lacking of practicing with native speakers.
* People may have difficulty in time scheduling when participating in language centers.
* A person may have difficulty if he/she want to learn an uncommon language.
* A person who is fluent on his/her mother tongue does not have a chance to teach if he/she wants.

## Proposed Solution

The proposed solution is to build a mobile application as a method for those who want to learn or teach languages, especially who want to focus on speaking and listening.

* 1. Featured functions

Web application:

* An UI for Staff to manage user
* A UI for Staff to manage teacher sign up request.

Mobile application:

* Register time slots: users who want to teach can register his/her available time slots so that other users can register to learn
* Course Tool: a simple tool helps teachers compose a course structure
* Learning booking: users who want to learn can book an available slot from teachers base on their demands
  1. Advantages
* Users can practice speaking effectively
* Learning time is flexible for both teacher and learner
* As long as there are people willing to teach, any language is available
  1. Disadvantages
* Teachers’ quality is not strictly controlled
* Heavily depends on mobile device and Internet connection

# Software Project Management Plan

## Problem Definition

### Name of this Capstone Project

* Official name: Mobile Based Language Learning Application
* Vietnamese name: Ứng Dụng Học Ngôn Ngữ Trên Thiết Bị Di Động
* Abbreviation: MOLA

### Problem Abstract

* There are certain limits in learning languages in traditional methods, such as learners have to take lessons in specific time and place. Interactions with native speakers in practicing is also a concern for a large number of language learners. The solution should be able to handle those matters.
* The system provide a mobile application that support learning languages as well as resolve different related demands. The application helps learners connecting with native teachers via video calls. Using scheduling functions, learners can select learning time as their wishes. Learners can learn at home, at coffee shop or anywhere with Internet connection. The system also provide other supporting tools, such as learn-time dictionary or suggesting suitable courses for learners.
* The system provide option for people to become teachers in the system. This will help the community using the system grows. As a result, users will have more choices when learning, as well as an earning method when teaching.

### Project Overview

#### Current situation

Below are the problems encountered in this project:

* New technique: some team members are new to the techniques used in the project. The team need an amount of time to get familiar with those techniques.
* Fix scope: the scope of the project is not simple to determine. There are some factors that may affect and the team have to decide scope correctly.
* Absence of team members: team members can get sick or unexpected problems.

#### The Proposed System

The system has essential features for the mobile application, such as managing courses, registering learning time slots and making learn-time video calls. The application will contain core business features of the system.

The system also has a web portal for administration activities. The web will contain user-related functions.

The system will maintain a high available web server with APIs to work 24/7 for responding requests from all over the world. The web server will be the main data handler.

We assign responsibility in vertical to make sure if any member in this problem cannot continue to work in our team there will be the least harmful to the project processes.

The system will have three sub-systems:

* A mobile application for users to perform learning and teaching activities
* A web portal for administration activities
* An API application to serve API for web portal and mobile application

##### Mobile Application

* For learner:
* Search courses
* Enroll in courses
* Register learning time slots
* Use learning supporting tools
* For teacher:
* Manage his/her courses
* Register available time slots
* For all users:
* Making learn-time video calls
* View profile
* Register to be teacher

##### Web Portal

The web portal is used mostly for administrators and staffs:

* Manage user accounts
* Approve/Reject teacher register

##### Web Server/ API Application

The server system takes responsibility to respond all the requests and also manages and processes data.

* Provide APIs for both Mobile Application and Web Portal
* Perform data processing
* Perform scheduled tasks

#### Boundaries of the System

This section suppose that users of the system has stable Internet connection while using.

* Allow teachers to create and manage his/her courses
* Allow teachers to register available time slots for teaching
* Allow learners to search courses, as well as view courses info and enroll in
* Allow learners to register learning time slots from available time slots of corresponding teacher
* Allow video call making between learner and teacher at the learning time
* Provide a course recommending mechanism for learners
* Supported languages is restricted at the first version

#### Future plans

Current system is concentrated on core business flow. Therefore, some supporting features are restricted adapted for the development team. These features may be expanded in the future:

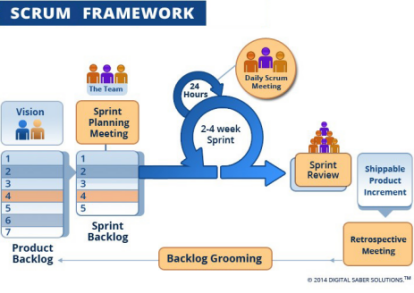
* Support most of available languages
* Optimize collaborative filtering for better recommendation

## Project organization

### Software Process Model

This project is developed using Scrum model – part of an agile framework for Software development project. Our team choose Scrum model because of the following reasons:

* Prototypes are delivered frequently for evaluation, usually weekly, rather than months.
* Take fewer risks when there is a change in requirement.
* All members must work together in order to avoid misunderstanding or miscommunication.
* Able to study new skills or knowledge at the same time as developing.



*Figure 1: Scrum Process*

(<http://www.digitalsaber.com/process/agile-scrum-framework/>)

### Roles and Responsibilities

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Full Name** | **Role in Group** | **Responsibilities** |
| 1 | Nguyễn Huy Hùng | Product owner | * Specify scope and user requirement. * Supervise the development progress. * Provide professional techniques and business analysis support. |
| 2 | Lê Ngọc Hiếu | Scrum master | * Managing process * Designing database * Clarifying requirements * Prepare documents * GUI design * Create test plan * Coding * Testing |
| 3 | Đỗ Hoàng Nguyên | Team member | * Designing database * Clarifying requirements * Prepare documents * GUI design * Coding * Testing |
| 4 | Nguyễn Nhật Quang | Team member | * Designing database * Clarifying requirements * Prepare documents * GUI design * Coding * Testing |
| 5 | Trương Thanh Lâm | Team member | * Designing database * Clarifying requirements * Prepare documents * GUI design * Coding * Testing |
|  | | | |

*Table 2: Roles and Responsibilities*

### Tools and Techniques

|  |  |
| --- | --- |
|  | Tool/Technique |
| Mobile Application | Android SDK, React Native, JavaScript, CSS |
| Web Portal | Play framework, Java, HTML 5 |
| Back-end | Play framework, Java, JPA |
| Temporary Storage | Redis |

*Table 3: Tools and Techniques*

## Project Management Plan

### Product Backlog

Refer to Appendix

### Sprint Backlog

Refer to Appendix

### Deliverables

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Deliverable** | **Deliverable date** | **Deliverable location** | **Note** |
| 1 | Introduction Document, Task list |  | FU - LMS | Report No. 1 |
| 2 | Software Project Management Plan |  | FU – LMS | Report No. 2 |
| 3 | Software Requirements Specification |  | FU – LMS | Report No. 3 |
| 4 | Software Design Description |  | FU – LMS | Report No. 4 |
| 5 | Software Test Documentation Guide Implementation (Coding) |  | FU – LMS | Report No. 5 |
| 6 | Software User’s Manual |  | FU - LMS | Report No. 6 |
|  | | | | |

*Table 4: Deliverables*

* For each Sprint, deliverables are potentially shippable products, which can be a part of document or prototype implemented based on the project’s core flow without any constraints.
* Each Sprint has a fixed duration of one weeks.

### All Meeting Minutes

All meeting documents could be found where

# Software Requirement Specifications

## User Requirement Specification

### 1.1 Guest Requirement

Guest is a person who does not have access to the system. User can use some function in the system. These are some function user can use:

* + - * + Sign in
        + Sign up

### 1.2 Learner Requirement

* + - * + Edit Profile
        + Search Course
        + View Teacher Profile
        + View Course Info
        + Register Course
        + Book Timeslot
        + Video Call
        + Chat
        + Rating Teacher
        + View Schedule
        + Sign up Teacher
        + Cancel Request

### 1.3 Teacher Requirement

Learner register to become teacher, teacher have all function of learner, and:

* + - * + Course Managements: CRUD course
        + Set Free Time Slot
        + Receive Request

### 1.4 Staff Requirement

Staff is person who review teacher’s sign up request

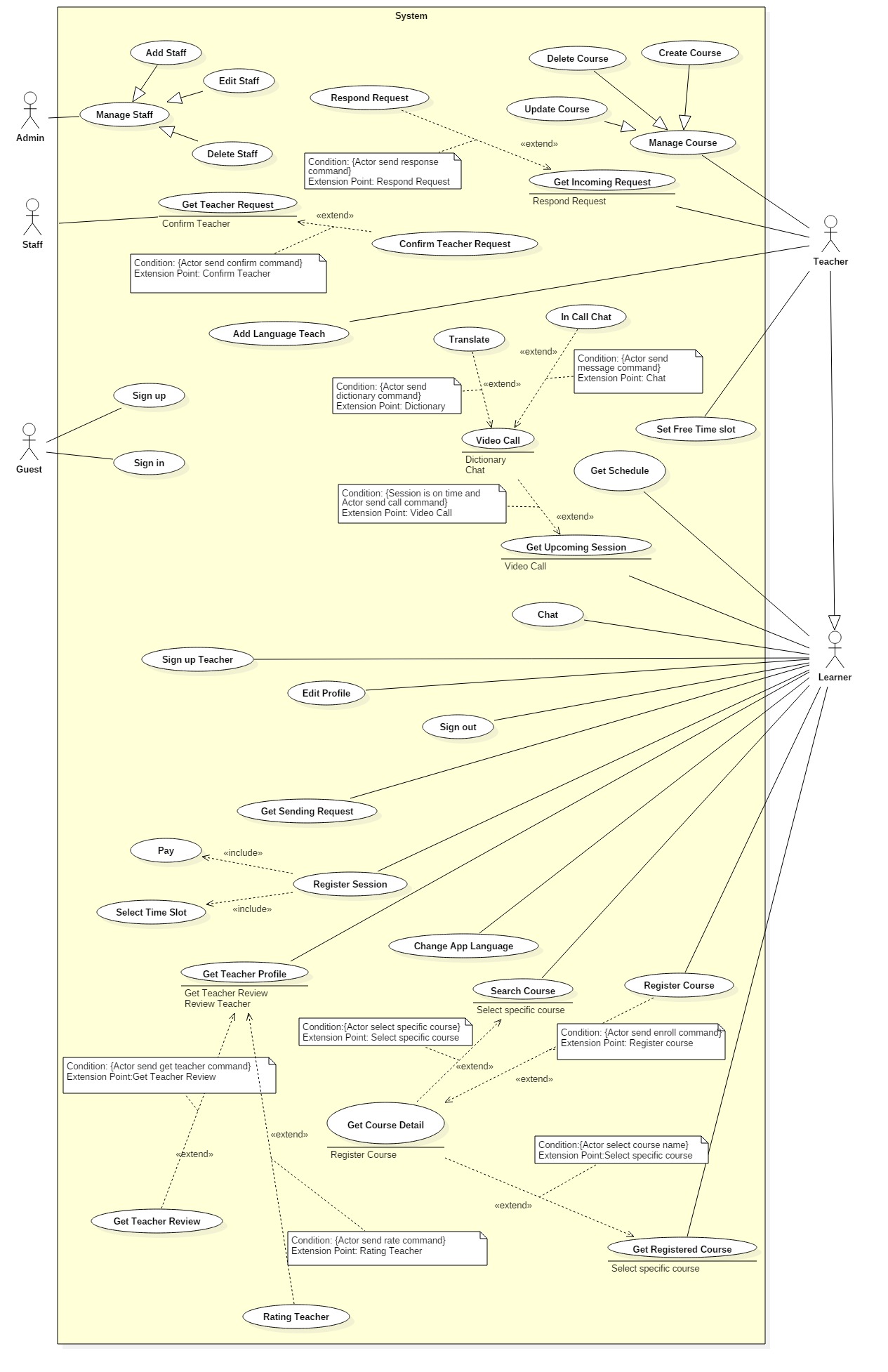
* + - * + Accept/ Decline teacher’s sign up request

### Admin Requirement

* Manage Staff:
  + CRUD staff

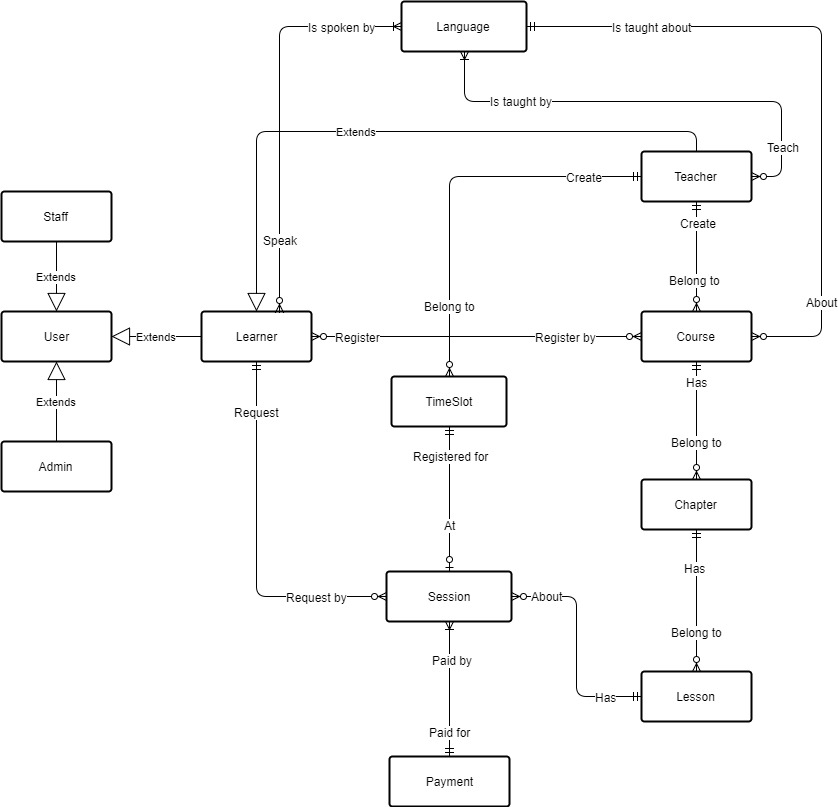
## System Requirement Specification

### System Overview Use Case



*Figure 2: Use case overview*

## Conceptual Diagram



*Figure 3: Conceptual Diagram*

**Data Dictionary**

|  |  |
| --- | --- |
| **Entity Name** | **Description** |
| User | Contains information of all users in the system. |
| Admin | Person who manages the system. |
| Staff | Person who review teacher request |
| Learner | Person who login to the system to learn language |
| Teacher | Person who login to the system to teach language |
| Language | Contains all language system support |
| Course | Contains information of all course in the system. |
| Chapter | Contains information of all chapter in the system. |
| Lesson | Contains information of all lesson in the system. |
| Time Slot | Contains information of free time slot of the teacher |
| Session | Contains information of session learner request to learn |
| Payment | Contains all payment method |

*Table 5: Conceptual Diagram Data Dictionary*

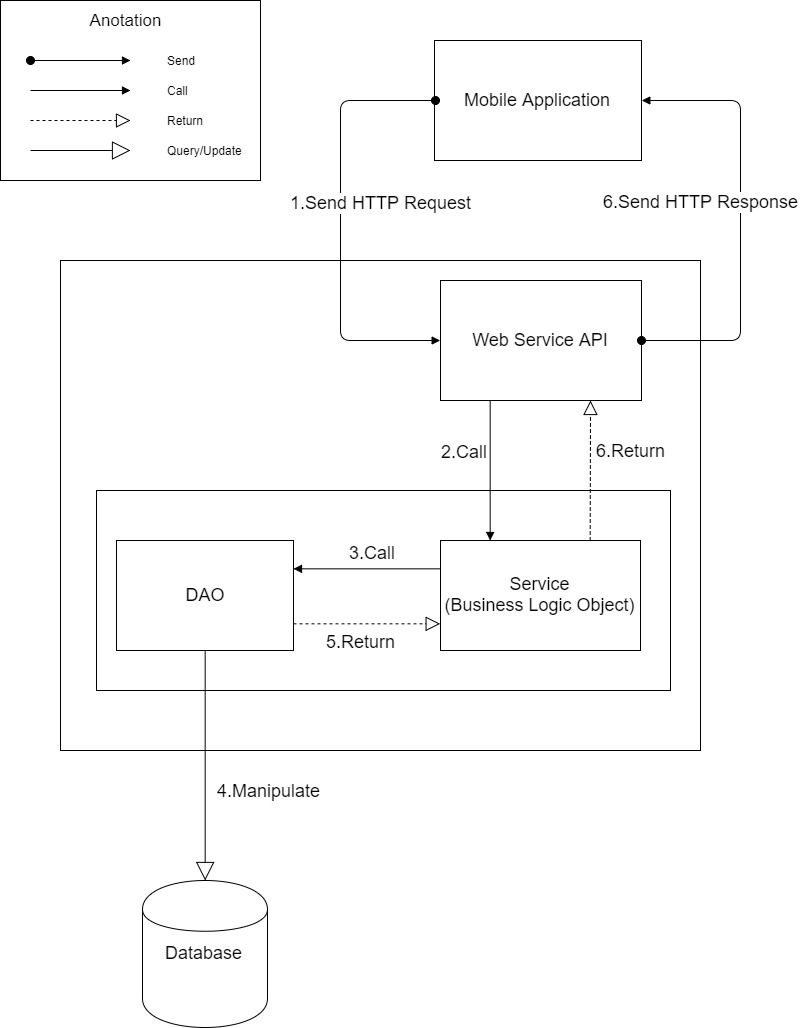
# Software Design Description

## Design Overview

* The architectural design describes the overall architecture of the system and the architecture of each main component and subsystem.
* The detailed design describes static and dynamic structure for each component and functions. It includes class diagrams, class explanations and sequence diagrams for each use cases.
* The database design describes the relationships between entities and details of each entity.
* Document overview:
* Section 1: gives an overall description of the system architecture design.
* Section 2: gives component diagrams that describe the connection and integration of the system.
* Section 3: gives the detail design description, which includes class diagram, class explanation, and sequence diagram to details the application functions.
* Section 4: gives the interface design description, which includes component interface, web application interface and mobile application design.
* Section 5: describe a fully attributed Entity Relationship Diagram.
* Section 6: describe the algorithms that apply to the system.

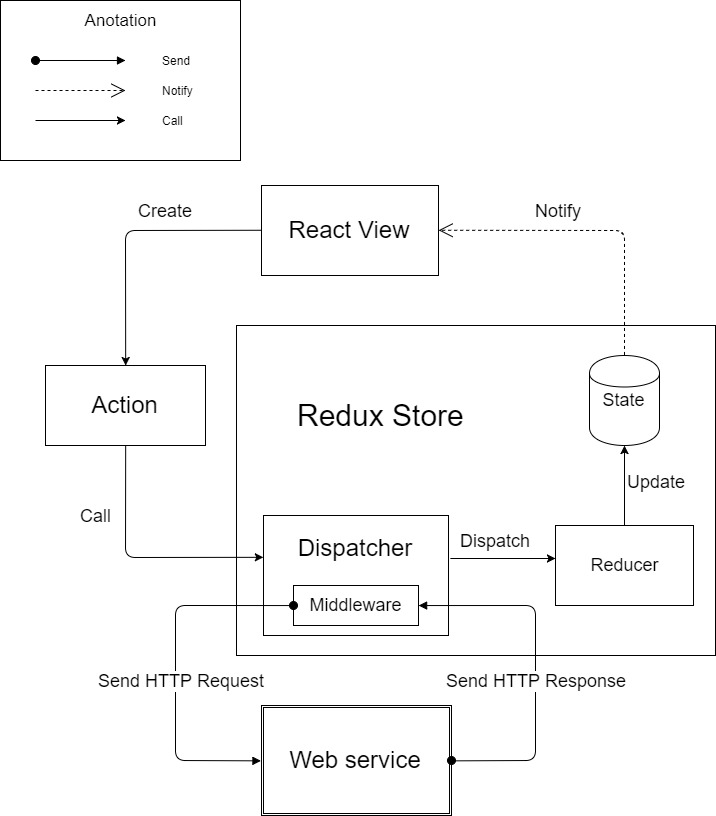
## System Architectural Design

### **Web service architecture description**

****

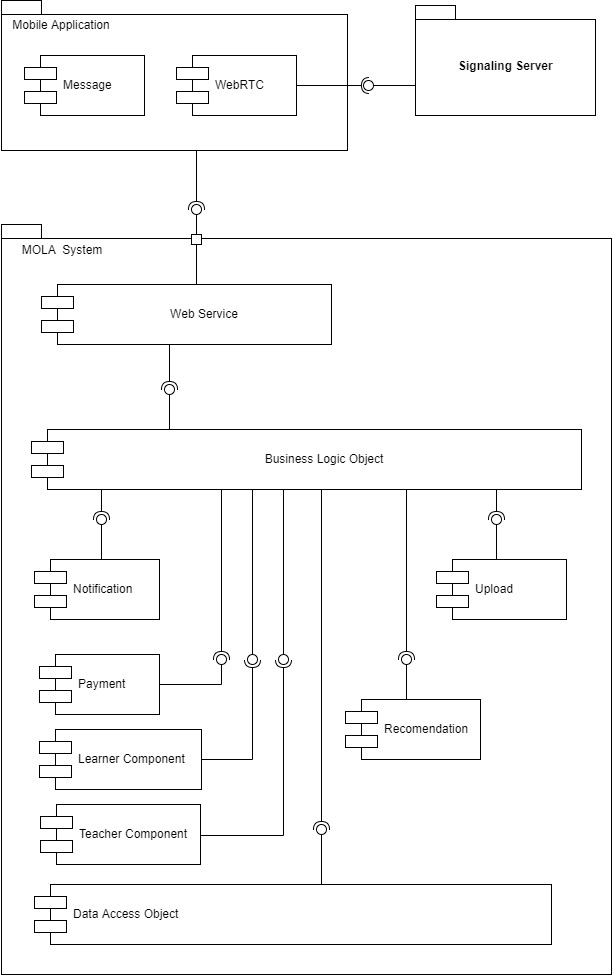
*Figure 4: Web service Architecture*

### Mobile Application Architecture Description



*Figure 5: Mobile Application Architecture*

## Component Diagram

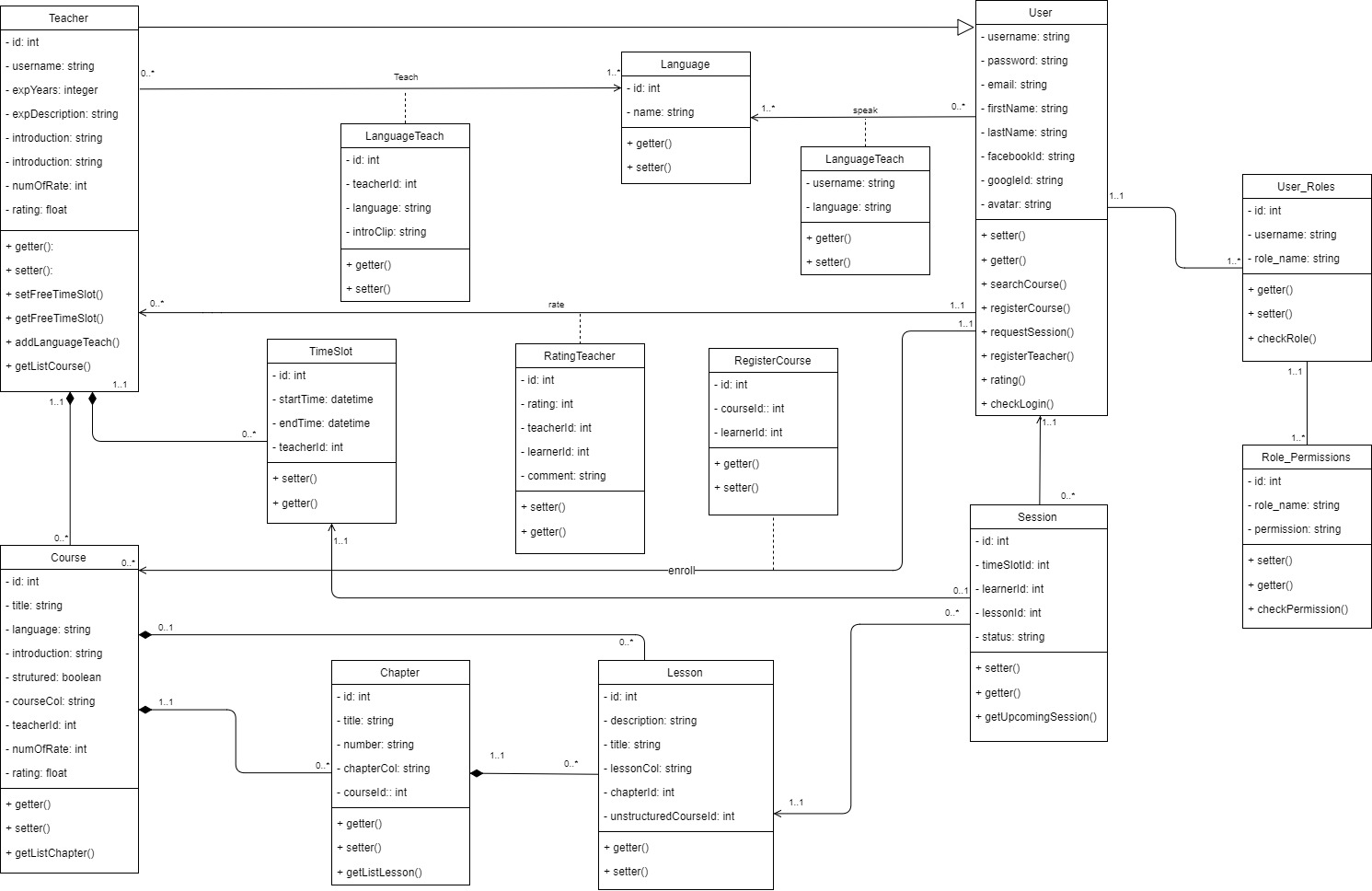


*Figure 6: Component Diagram*

|  |  |
| --- | --- |
| **Name** | **Description** |
| MOLA System | Interaction Learning Language system |
| Web Application | Web application package: View, Controller |
| Mobile Application | Mobile application package |
| Web Service | Provide API for mobile applications to interact with the system. |
| Business Logic Object | Common objects to handle domain business operations |
| Data Access Object | Component to handle interaction between the system and  database |
| Payment Component | Component to handle payment process |
| PayPal | Handle payment process with PayPal API |
| Firebase | External Component to send notification to user and allow user to send message to each other. |
| Notification Component | Component to interaction with Firebase to send notification to user. |
| Active Record | Abstract data layer to interact with database system |
| Database System | Where application data is manipulated and stored |
| Scheduler | Run background job and send message to other system |
| Mail System | External Component to send email |
| Cloud Message | External Component to send notification to user |
| *Table 6: Component Diagram Dictionary* | |

## Detailed Description

### Class Diagram



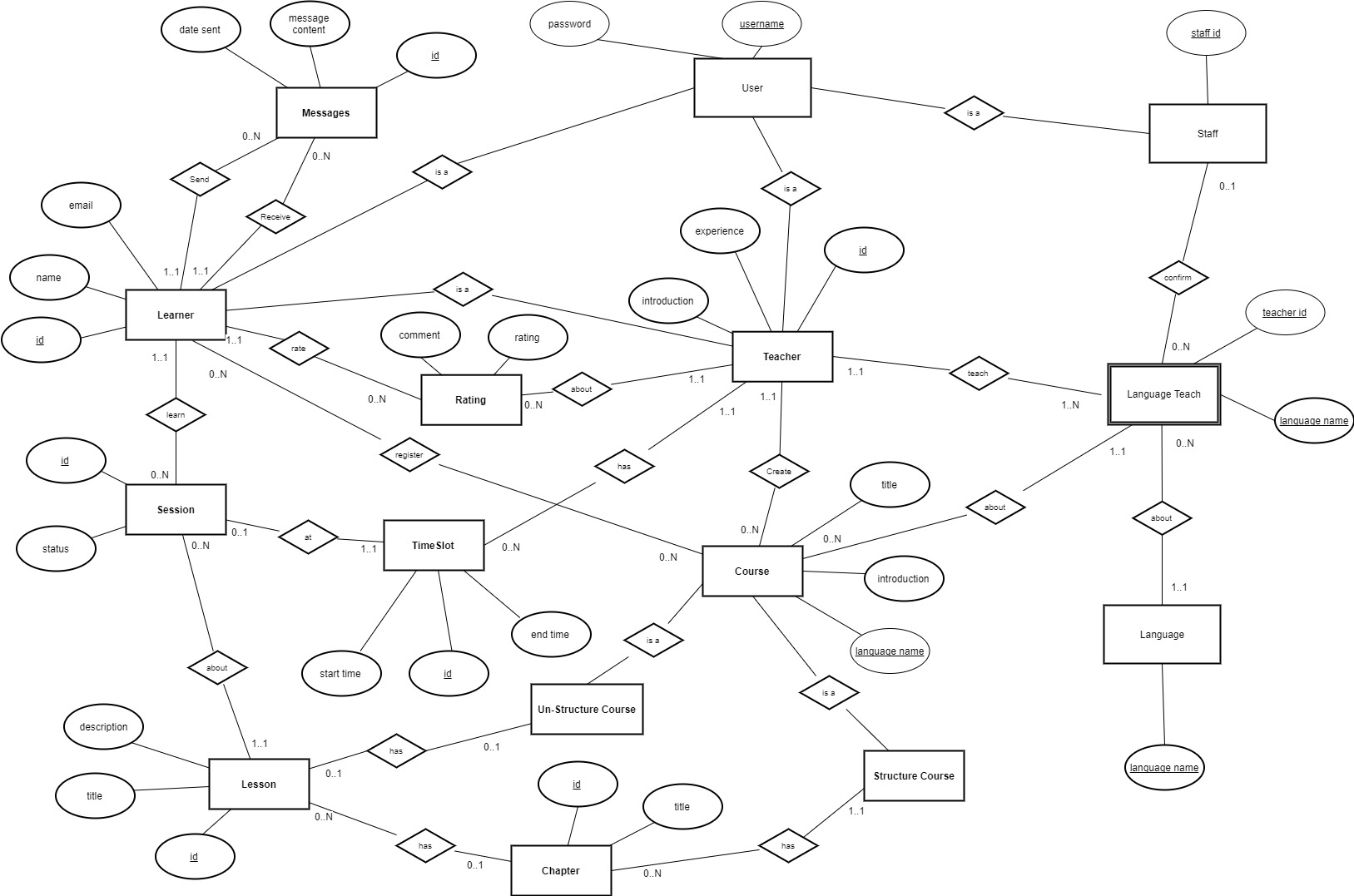
*Figure 7: Class Diagram*

|  |  |  |
| --- | --- | --- |
| **Class name** | **Mapping column with conceptual diagram** | **Description** |
| Teacher Entity | Teacher | Person who login to the system to teach language |
| Course Entity | Course | Contains information of all course in the system. |
| Chapter Entity | Chapter | Contains information of all chapter in the system. |
| Lesson Entity | Lesson | Contains information of all lesson in the system. |
| User Entity | User | Person who login to the system to learn language |
| User Role Entity | N/A | Contains information of role of the user in the system |
| Role Permission Entity | N/A | Contains permission of role in the system |
| Session Entity | Session | Contains information of all session in the system |
| Time Slot Entity | Time Slot | Contains information of all time slot in the system |
| Rating Teacher Entity | N/A | Contains information of rating of teacher in the system |
| Register Course Entity | N/A | Contains information of registered course in the system |
| Language Teach Entity | N/A | Contains language that the teacher teach |
| Language Speak Entity | N/A | Contains language that the user speak |
| Language Learn Entity | N/A | Contains language that the user learn |
| Payment Entity | Payment | Contains information of payment in the system |
|  | | |

*Table 7: Class Diagram Dictionary*

## Database Design

### Entity Relationship Diagram (ERD)



*Figure 8: Entity Relationship Diagram*

### Data Dictionary

|  |  |
| --- | --- |
| **Entity Data Dictionary: describe contents of all entities** | |
| **Entity Name** | **Description** |
| User | Contains information of all users in the system. |
| Teacher | Contains information of all teachers in the system. |
| Course | Contains information of all courses in the system. |
| Chapter | Contains information of chapter in the system. |
| Lesson | Contains information of lesson in the system. |
| Time Slot | Contains information of time slot in the system. |
| Session | Contains information of session in the system. |
| Learner | Contains information of all learner in the system. |
| Rating | Contains information of rating teacher the system. |
| Staff | Contains information of all staffs in the system. |
| Language | Contains information of all languages in the system. |
| Message | Contains information of all messages in the system. |
| Language Teach | Contains information of language teacher teach |

*Table 8: ERD dictionary*

## Algorithms

### Recommendation

#### Definition

Recommendation is the way to introduce courses to learners. When learners are not sure which course to start or continue with, they can refer to those suggestions.

For different situations, different recommendation methods should be applied.

#### Define Problem

There are different situations that need recommendation. Each of those requires a certain method.

We also have to ensure that new courses and courses from newly registered teachers have opportunity to approach potential learners.

#### Solution

Recommendation algorithms are introduced in the book *Programming Collective Intelligence* by Toby Segaran, published on August 2007 by O’Reilly.

We build the dataset for recommendation by considering all courses interacted by each learner.

The score of a learner to a course is determined by a formula with the following elements:

* Percentage of completed lessons by the learner on the course. The percentage point is scaled linearly from 0 to 5, with 0% results in 0 and 100% results in 5.
* Average rating of the teacher who created the course. This element always has value between 0 and 5.
* Number of registration on the course per week. The value is scaled linearly from 0 to 5, with 0 results in 0 and 25 or more results in 5.
* Price of the course. The price is scaled linearly from 0 to 5, with 20 USD minute results in 0 and 0 USD (free course) results in 5.

Each of those elements has its own factor. After multiplying the element with its factor and sum them up, the value is divided by course existed time (in hours) so that newly created course will have high chance to approach learners. The final formula is:

For a new learner or a learner who interacts with few courses, we make a separated dataset. We consider all courses in languages that learner is learning. The score for each course has all of the above elements except for percentage of completed lessons. The factor for each element will be different, also. The final formula for this kind of learner is:

We use the following methods for specific situation:

1. When a learner looks for courses, we use User-based Recommendation for suggesting.
2. When a learner has interest in a specific course, we suggest similar courses using Item Comparison Dataset.
3. We use the new learner dataset directly to suggest for this kind of learners.

#### Complexity

* The complexity for building standard recommendation dataset is ***N4***.
* The complexity for building comparison dataset is ***N2***.
* The complexity for User-based Recommendation is ***N2***.
* The complexity for building new user dataset is ***N2***.

In total, the complexity for this algorithm is ***N4***.

#### Example

In this example, there are following English courses in the system:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Course | Teacher Rating | Registration | Existed Time | Unit Price |
| 1 | 4.0 | 10 | 1 week | 4 |
| 2 | 4.0 | 50 | 2 weeks | 2 |
| 3 | 3.5 | 10 | 4 weeks | 12 |
| 4 | 3.5 | 20 | 2 weeks | 4 |
| 5 | 3.5 | 35 | 1 week | 0 |
| 6 | 2.0 | 50 | 2 weeks | 0 |
| 7 | 2.0 | 50 | 4 weeks | 2 |
| 8 | 2.0 | 20 | 2 weeks | 6 |
| 9 | 4.5 | 150 | 10 weeks | 8 |
| 10 | 0 | 0 | 1 day | 4 |

There are also following registration:

|  |  |  |
| --- | --- | --- |
| Learner | Course | % Complete |
| 1 | 1 | 60 |
| 1 | 2 | 60 |
| 1 | 3 | 10 |
| 1 | 5 | 70 |
| 1 | 8 | 20 |
| 1 | 9 | 80 |
| 2 | 1 | 20 |
| 2 | 2 | 30 |
| 2 | 4 | 60 |
| 2 | 7 | 50 |
| 2 | 9 | 90 |
| 3 | 1 | 60 |
| 3 | 3 | 80 |
| 3 | 4 | 50 |
| 3 | 6 | 90 |
| 3 | 7 | 30 |
| 3 | 8 | 80 |
| 4 | 2 | 40 |
| 4 | 4 | 30 |
| 4 | 5 | 60 |
| 4 | 7 | 70 |
| 4 | 8 | 40 |
| 4 | 9 | 50 |
| 5 | 3 | 40 |
| 5 | 4 | 40 |
| 5 | 5 | 30 |
| 5 | 7 | 100 |
| 6 | 1 | 60 |
| 6 | 3 | 40 |
| 6 | 5 | 70 |
| 6 | 8 | 90 |
| 6 | 9 | 80 |
| 7 | 1 | 50 |
| 7 | 2 | 40 |
| 7 | 3 | 100 |
| 7 | 4 | 20 |
| 7 | 6 | 40 |
| 7 | 7 | 80 |
| 7 | 8 | 20 |

After applying the formula, the dataset is built as below:

|  |  |  |
| --- | --- | --- |
| Learner | Course | Score |
| 1 | 1 | 3.2 |
| 1 | 2 | 1.925 |
| 1 | 3 | 0.3875 |
| 1 | 5 | 3.95 |
| 1 | 8 | 0.875 |
| 1 | 9 | 0.385 |
| 2 | 1 | 2.4 |
| 2 | 2 | 1.625 |
| 2 | 4 | 1.525 |
| 2 | 7 | 0.6375 |
| 2 | 9 | 0.405 |
| 3 | 1 | 3.2 |
| 3 | 3 | 0.7375 |
| 3 | 4 | 1.425 |
| 3 | 6 | 1.95 |
| 3 | 7 | 0.5375 |
| 3 | 8 | 1.475 |
| 4 | 2 | 1.725 |
| 4 | 4 | 1.225 |
| 4 | 5 | 3.75 |
| 4 | 7 | 0.7375 |
| 4 | 8 | 1.075 |
| 4 | 9 | 0.325 |
| 5 | 3 | 0.5375 |
| 5 | 4 | 1.325 |
| 5 | 5 | 3.15 |
| 5 | 7 | 0.8875 |
| 6 | 1 | 3.2 |
| 6 | 3 | 0.5375 |
| 6 | 5 | 3.95 |
| 6 | 8 | 1.575 |
| 6 | 9 | 0.385 |
| 7 | 1 | 3 |
| 7 | 2 | 1.725 |
| 7 | 3 | 0.8375 |
| 7 | 4 | 1.125 |
| 7 | 6 | 1.45 |
| 7 | 7 | 0.7875 |
| 7 | 8 | 0.875 |

From the dataset, top user-based recommendations for each learner will be:

|  |  |
| --- | --- |
| Learner | Courses |
| 1 | 6, 4 |
| 2 | 5, 6 |
| 3 | 5, 2 |
| 4 | 1, 6 |
| 5 | 1, 2 |
| 6 | 2, 6 |
| 7 | 5, 9 |

Also from the dataset, Comparison Dataset is produced as below:

|  |  |
| --- | --- |
| Course | Similar Courses |
| 1 | 5, 2, 6 |
| 2 | 6, 4, 8 |
| 3 | 9, 7, 4 |
| 4 | 8, 6, 2 |
| 5 | 1, 2, 4 |
| 6 | 2, 4, 8 |
| 7 | 3, 9, 8 |
| 8 | 4, 6, 7 |
| 9 | 3, 7, 4 |

When a learner view detailed information of a specific course, its similar courses will be recommended.

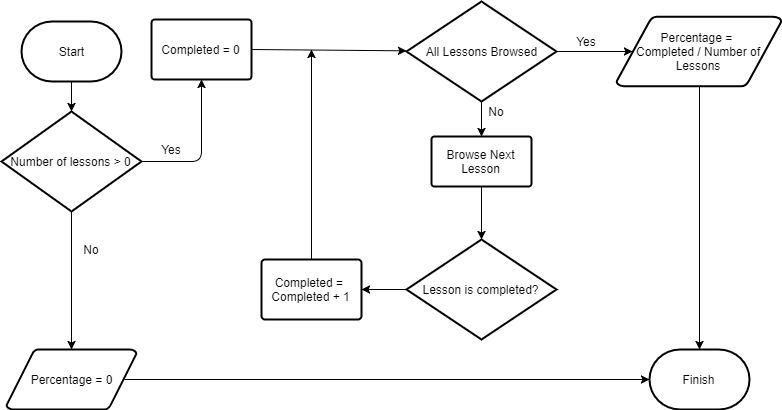
Beside, English dataset for new learner can be generated applying the formula:

|  |  |
| --- | --- |
| Course | Score |
| 1 | 3.2 |
| 2 | 2.225 |
| 3 | 0.5375 |
| 4 | 1.475 |
| 5 | 4.25 |
| 6 | 1.75 |
| 7 | 0.6125 |
| 8 | 1.075 |
| 9 | 0.375 |
| 10 | 2.8 |

Based on new learner dataset, for a new learner, courses No. 5, 1 and 10 will be recommended.

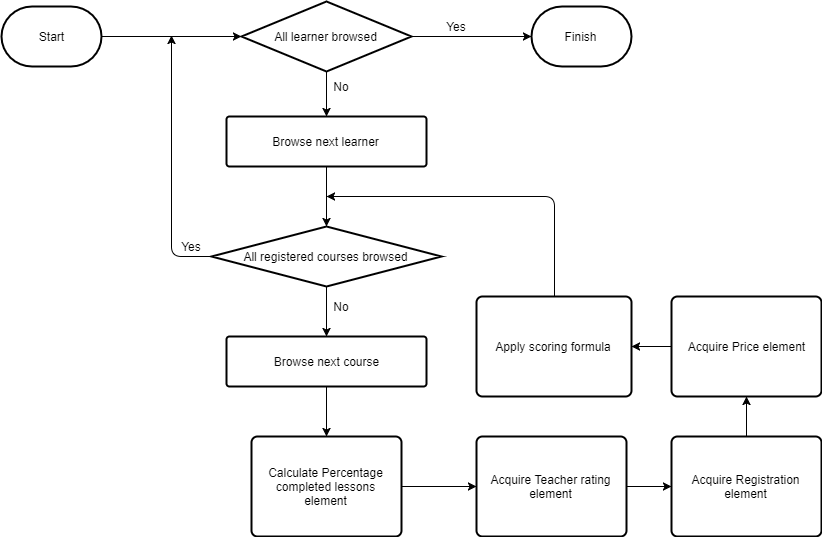
#### Flowchart

##### Percentage of completed lessons



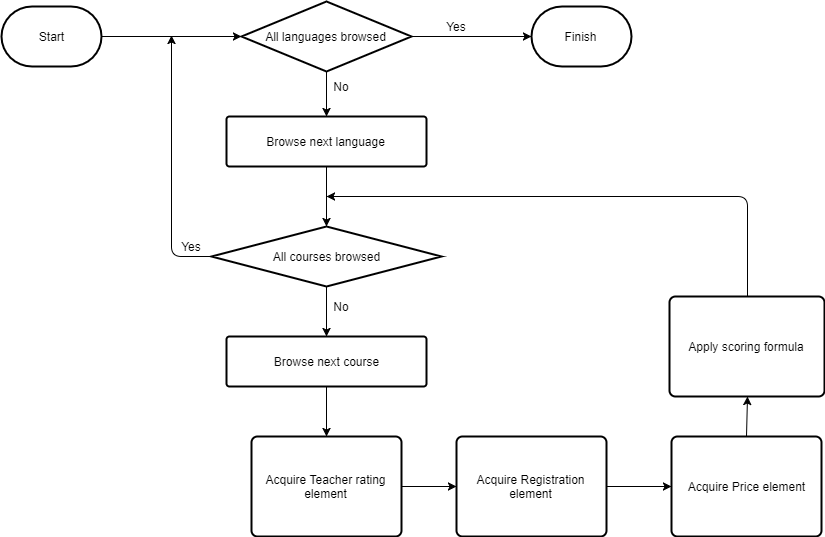
*Figure 9: Percentage of completed lessons flow chart*

##### Build Dataset



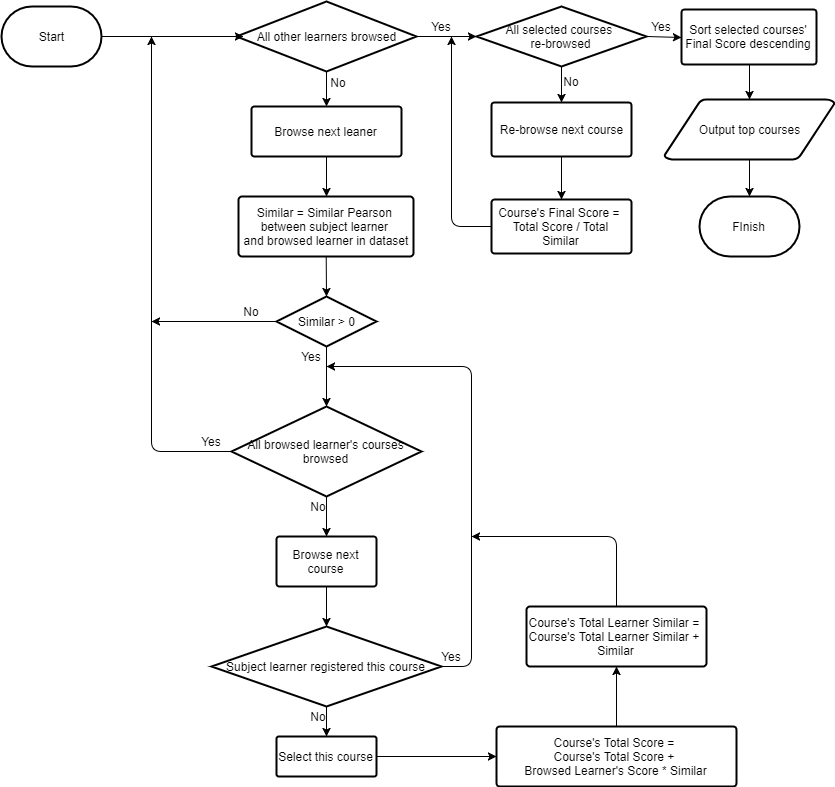
*Figure 10: Build Dataset Flow Chart*

##### Build New User Dataset



*Figure 11: Build New User Dataset Flow Chart*

##### User-Based Recommendation



*Figure 12: User- based Recommendation Flow Chart*

# Tasksheet

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Product Deliverables** | **Task** | **HieuLN** | **NguyenDH** | **QuangNN** | **LamTT** | **Unit** | **Size** |
| 1 | Report1 - Introduction | Project Information | **O** |  |  |  |  | 1 |
| Introduction |  |  |  | **O** |  | 1 |
| Current Situation |  |  | **O** | **O** |  | 1 |
| Problem Definition | **O** |  |  |  |  | 1 |
| Proposed Solution |  | **O** |  |  |  | 1 |
| Functional Requirements | **O** | **O** |  |  |  | 1 |
| Role and Responsibility | **O** |  |  |  |  | 1 |
| Review and merge document | **O** |  |  |  |  | 1 |
| 2 | Report2- Software Project Management Plan | Problem Definition |  |  | **O** |  |  | 1 |
| Project Organization | **O** |  |  |  |  | 1 |
| Project Management Plan | **O** | **O** |  |  |  | 1 |
| Coding Convention |  |  |  | **O** |  | 1 |
| Review and merge document | **O** |  |  |  |  | 1 |
| 3 | Report 3- Software Requirement Specification | **User Requirement Specification** |  |  | **O** |  |  | 1 |
| **System Requirement Specification** |  |  |  |  |  |  |
| **External Interface Requirements** |  | **O** |  |  |  | 1 |
| **Functional Requirement** | **O** |  |  |  |  | 1 |
| **System Overview Usecase** | **O** |  |  |  |  | 3 |
| **List of Usecase** |  |  |  |  |  |  |
| **Admin** |  |  |  |  |  |  |
| Add Staff |  |  | **O** |  |  | 1 |
| Edit Staff |  |  | **O** |  |  | 1 |
| Delete Staff |  |  | **O** |  |  | 1 |
| **Guest** |  |  |  |  |  |  |
| Sign Up |  |  | **O** |  |  | 1 |
| Sign In |  |  | **O** |  |  | 1 |
| **Learner** |  |  |  |  |  |  |
| Sign Out |  |  | **O** |  |  | 1 |
| Edit Profile |  | **O** |  |  |  | 1 |
| Get List Conversation |  | **O** |  |  |  | 1 |
| Chat |  | **O** |  |  |  | 1 |
| Get Schedule |  | **O** |  |  |  | 1 |
| Get Upcoming Session |  | **O** |  |  |  | 1 |
| Call Video | **O** |  |  |  |  | 1 |
| Translate | **O** |  |  |  |  | 1 |
| In Call Chat | **O** |  |  |  |  | 1 |
| Get Sending Request | **O** |  |  |  |  | 1 |
| Sign Up Teacher | **O** |  |  |  |  | 1 |
| Register Session | **O** |  |  |  |  | 1 |
| Get Teacher Profile |  |  |  | **O** |  | 1 |
| Get Teacher Review |  |  |  | **O** |  | 1 |
| Rating Teacher |  |  |  | **O** |  | 1 |
| Change App Language |  |  |  | **O** |  | 1 |
| Search Course |  |  |  | **O** |  | 1 |
| Get Course Detail |  |  |  | **O** |  | 1 |
| Register Course |  |  |  | **O** |  | 1 |
| Get Registered Course |  |  |  | **O** |  | 1 |
| **Teacher** |  |  |  |  |  |  |
| Create Course |  |  | **O** |  |  | 1 |
| Update Course |  |  | **O** |  |  | 1 |
| Delete Course |  |  | **O** |  |  | 1 |
| Get Incoming Request |  |  | **O** |  |  | 1 |
| Responde Request |  |  | **O** |  |  | 1 |
| Set Free Timeslot |  |  | **O** |  |  | 1 |
| Add Language Teach |  |  | **O** |  |  | 1 |
| **Software System Attribute** |  |  |  |  |  |  |
| Usability | **O** |  |  |  |  | 1 |
| Reliability | **O** |  |  |  |  | 1 |
| Availability |  | **O** |  |  |  | 1 |
| Secutiry |  | **O** |  |  |  | 1 |
| Maintainability |  |  | **O** |  |  | 1 |
| Portability |  |  |  | **O** |  | 1 |
| Performance |  |  |  | **O** |  | 1 |
| **Conceptual Diagram** | **O** |  |  |  |  | 3 |
| Review and merge document | **O** |  |  |  |  | 1 |
| 4 | Report 4- Software Design Description | Design Overview | **O** |  |  |  |  | 1 |
| System Architectural Design | **O** |  |  |  |  | 3 |
| Component Diagram | **O** |  |  |  |  | 2 |
| Class Diagram | **O** |  | **O** |  |  | 3 |
| Class Diagram Dictionary |  |  | **O** |  |  | 1 |
| Class Diagram Explanation |  |  | **O** |  |  | 1 |
| **Interaction Diagram** |  |  |  |  |  |  |
| **Activity Diagram** |  |  |  |  |  |  |
| Login |  |  |  | **O** |  | 1 |
| Register Teacher |  |  |  | **O** |  | 1 |
| Register Course |  | **O** |  |  |  | 1 |
| Request Session |  | **O** |  |  |  | 1 |
| Create Course |  | **O** |  |  |  | 1 |
| Set Time Slot |  | **O** |  |  |  | 1 |
| **Sequence Diagram** |  |  |  |  |  |  |
| Register Teacher | **O** |  |  |  |  | 1 |
| Register Course | **O** |  |  |  |  | 1 |
| Request Session | **O** |  |  |  |  | 1 |
| Set Time Slot | **O** |  |  |  |  | 1 |
| **Interface** |  |  |  |  |  |  |
| Component Interface | **O** |  | **O** |  |  | 1 |
| User Interface Design | **O** | **O** | **O** | **O** |  | 1 |
| **Database Design** |  |  |  |  |  |  |
| Entity Relationship Diagram | **O** |  |  |  |  | 3 |
| Data dictionary |  |  | **O** |  |  | 1 |
| **Algorithms** |  |  |  |  |  |  |
| Recommendation |  |  |  | **O** |  | 5 |
| Review and merge document | **O** |  |  |  |  | 1 |
| 5 | Report 5 - Software Implementation and Test Document | **Introduction** |  |  |  |  |  |  |
| Overview | **O** |  | **O** |  |  | 1 |
| Test Approach |  | **O** |  |  |  | 1 |
| **Database Relationship Diagram** |  |  |  |  |  |  |
| Physical Diagram | **O** | **O** |  |  |  | 3 |
| Data dictionary |  |  | **O** | **O** |  | 1 |
| **Mobile Implementation** |  |  |  |  |  |  |
| **Learner** |  |  |  |  |  |  |
| Sign Out |  |  | **O** |  |  | 1 |
| Edit Profile |  |  | **O** |  |  | 1 |
| Get List Conversation |  |  |  | **O** |  | 1 |
| Chat |  | **O** |  |  |  | 1 |
| Get Schedule |  | **O** |  |  |  | 1 |
| Get Upcoming Session |  | **O** |  |  |  | 1 |
| Call Video |  | **O** |  |  |  | 1 |
| Translate | **O** |  |  |  |  | 1 |
| In Call Chat |  |  |  | **O** |  | 1 |
| Get Sending Request |  |  | **O** |  |  | 1 |
| Sign Up Teacher | **O** |  |  |  |  | 1 |
| Register Session | **O** |  |  |  |  | 1 |
| Get Teacher Profile |  |  | **O** |  |  | 1 |
| Get Teacher Review |  |  | **O** |  |  | 1 |
| Rating Teacher |  |  | **O** |  |  | 1 |
| Change App Language |  |  |  | **O** |  | 1 |
| Search Course |  | **O** |  |  |  | 1 |
| Get Course Detail |  |  | **O** |  |  | 1 |
| Register Course | **O** |  |  |  |  | 1 |
| Get Registered Course | **O** |  |  |  |  | 1 |
| **Teacher** |  |  |  |  |  |  |
| Create Course |  |  | **O** |  |  | 1 |
| Update Course |  |  | **O** |  |  | 1 |
| Delete Course |  |  | **O** |  |  | 1 |
| Get Incoming Request | **O** |  |  |  |  | 1 |
| Responde Request | **O** |  |  |  |  | 1 |
| Set Free Timeslot |  | **O** |  |  |  | 1 |
| Add Language Teach |  |  |  | **O** |  | 1 |
| **Web Application Implementation** |  |  |  |  |  |  |
| **Staff** |  |  |  |  |  |  |
| Login |  |  |  | **O** |  | 1 |
| Logout |  |  |  | **O** |  | 1 |
| Responde Request |  |  | **O** |  |  | 1 |
| **Admin** |  |  |  |  |  |  |
| Create Account | **O** |  |  |  |  | 1 |
| Update Account | **O** |  |  |  |  | 1 |
| Deactive Account | **O** |  |  |  |  | 1 |
| **API Implementation** |  |  |  |  |  |  |
| **Learner** |  |  |  |  |  |  |
| Edit Profile |  | **O** |  |  |  | 1 |
| Get List Conversation |  | **O** |  |  |  | 1 |
| Chat |  | **O** |  |  |  | 1 |
| Get Schedule |  | **O** |  |  |  | 1 |
| Get Upcoming Session | **O** |  |  |  |  | 1 |
| Call Video |  | **O** |  |  |  | 1 |
| Translate |  |  |  | **O** |  | 1 |
| In Call Chat |  | **O** |  |  |  | 1 |
| Get Sending Request | **O** |  |  |  |  | 1 |
| Sign Up Teacher | **O** |  |  |  |  | 1 |
| Register Session | **O** |  |  |  |  | 1 |
| Get Teacher Profile |  |  | **O** |  |  | 1 |
| Get Teacher Review |  |  | **O** |  |  | 1 |
| Rating Teacher |  |  | **O** |  |  | 1 |
| Change App Language |  |  |  | **O** |  | 1 |
| Search Course |  | **O** |  |  |  | 1 |
| Get Course Detail |  |  |  | **O** |  | 1 |
| Register Course |  |  |  | **O** |  | 1 |
| Get Registered Course |  |  |  | **O** |  | 1 |
| **Teacher** |  |  |  |  |  |  |
| Create Course |  |  | **O** |  |  | 1 |
| Update Course |  |  | **O** |  |  | 1 |
| Delete Course |  |  | **O** |  |  | 1 |
| Get Incoming Request | **O** |  |  |  |  | 1 |
| Responde Request |  |  |  | **O** |  | 1 |
| Set Free Timeslot |  | **O** |  |  |  | 1 |
| Add Language Teach | **O** |  |  |  |  | 1 |
| **Test plan** |  |  |  |  |  |  |
| Features to be tested |  |  | **O** |  |  | 1 |
| Features not to be tested |  |  | **O** |  |  | 1 |
| **System Testing Test Case** |  |  |  |  |  |  |
| Test Flow | **O** |  |  |  |  | 1 |
| **Test Case** |  |  |  |  |  |  |
| **Mobile application test cases** |  |  |  |  |  |  |
| **Guest** |  |  |  |  |  |  |
| Sign in | **O** |  |  |  |  | 1 |
| Sign up | **O** |  |  |  |  | 1 |
| **Learner** |  |  |  |  |  |  |
| Get upcoming session |  | **O** |  |  |  | 1 |
| Search Course |  | **O** |  |  |  | 1 |
| Get Course Details |  |  | **O** |  |  | 1 |
| Register Course |  | **O** |  |  |  | 1 |
| Select lesson |  |  |  | **O** |  | 1 |
| Select time slot |  |  | **O** |  |  | 1 |
| Payment |  | **O** |  |  |  | 1 |
| Video Call |  |  | **O** |  |  | 1 |
| Sign up teacher |  |  | **O** |  |  | 1 |
| **Teacher** |  |  |  |  |  |  |
| Set Free Timeslot | **O** |  |  |  |  | 1 |
| Create Course | **O** |  |  |  |  | 1 |
| Create Chapter |  |  | **O** |  |  | 1 |
| Create lesson |  | **O** |  |  |  | 1 |
| Respond Request |  |  |  | **O** |  | 1 |
| 6 | Report 6 - Software User's Manual | Review and merge document | **O** |  |  |  |  | 1 |
| Installation Guide | **O** | **O** |  |  |  | 1 |
| **User’s Guide** |  |  |  |  |  |  |
| ***Mobile Application*** |  |  |  |  |  |  |
| **Guest** |  |  |  |  |  |  |
| Sign In | **O** |  |  |  |  | 1 |
| Sign Up | **O** |  |  |  |  | 1 |
| **Learner** |  |  |  |  |  |  |
| Sign up teacher |  | **O** |  |  |  | 1 |
| Search Course |  | **O** |  |  |  | 1 |
| Enroll course | **O** |  |  |  |  | 1 |
| Request Time slot | **O** |  |  |  |  | 1 |
| Paid |  |  | **O** |  |  | 1 |
| Video Call |  |  | **O** |  |  | 1 |
| **Teacher** |  |  |  |  |  |  |
| Manage Course |  |  |  | **O** |  | 1 |
| Set Free Timeslot |  |  |  | **O** |  | 1 |
| Confirm request |  |  | **O** |  |  | 1 |
| Review and merge document | **O** |  |  |  |  | 1 |
|  |  |  |  |  |  |  |  |  |
| **Notes** |  |  |  |  |  |  |  |  |
| No. | Function Types | Function Point Size |  |  |  |  |  |  |
| 1 | Insert, Delete, Update, Simple Search, Login, Register, simple function or simple query on one table | 1 |  |  |  |  |  |  |
| 2 | Computed functions or joined query | 2 |  |  |  |  |  |  |
| 3 | Complex functions or multiple subquery | 3 |  |  |  |  |  |  |
| 4 | Constraint process and data integrity | 4 |  |  |  |  |  |  |
| 5 | Functions with implementing algorithms | 5 |  |  |  |  |  |  |

# Appendix

## UML Documentation of IBM

<http://www.ibm.com/developerworks/rational/library/769.html?ca=drs->

## React Native

<https://facebook.github.io/react-native/docs/getting-started.html>

## WebRTC

<https://webrtc.org/>

## Play Framework 2.5

<https://www.playframework.com/documentation/2.5.x/Home>

## Redis Cache

<https://redis.io/documentation>

## Firebase Cloud Message

<https://firebase.google.com/docs/cloud-messaging/>

## Firebase Realtime Database

<https://firebase.google.com/docs/database/>

## Dictionary API

<https://glosbe.com/>