IITB Summer Internship 2013
Software Design Description

edX Development

Principal Investigator
Prof. D. B. Phatak

Project In-charge
Mr. Nagesh Karmali

Project Mentors
Miss. Birundha M.
Miss. Firuza Aibara (PMO)
Mr. Prashant Kumar Singh
Mr. Srinu Kolokuluri
Mr. Vamsi Krishna

Project Team Members
Mr. Gagan Khanijau
Mr. Piyush Tiwari
Miss. Rupal Sharma
Miss. Sonam Gupta
Mr. Suhanth Boddu
Miss. Surabhi Prasad
Mr. Udayan Jain
Mr. Zeel Shah
Table of Contents
1. Introduction ................................................................. 4
   1.1 Purpose ............................................................... 4
   1.2 Scope ............................................................... 4
2. Definitions ............................................................... 5
3. References ............................................................... 5
4. Consideration for producing SDD ...................................... 6
   4.1 Software Life cycle .................................................. 6
   4.2 Purpose of SDD ...................................................... 7
5. Design description information content ............................... 7
   5.1 Introduction .......................................................... 7
   5.2 edX-platform ......................................................... 8
      5.2.1 Identification .................................................. 8
      5.2.2 Type ........................................................... 8
      5.2.3 Purpose ........................................................ 8
      5.2.4 Function ....................................................... 8
      5.2.5 Sub-modules .................................................. 8
      5.2.6 Dependencies .................................................. 9
   5.3 Codejail .............................................................. 9
      5.3.1 Identification .................................................. 9
      5.3.2 Type ........................................................... 9
      5.3.3 Purpose ........................................................ 9
      5.3.4 Function ....................................................... 10
      5.3.5 Sub-modules .................................................. 10
      5.3.6 Dependencies .................................................. 10
   5.4 Discussion forum ................................................... 10
      5.4.1 Identification .................................................. 10
      5.4.2 Type ........................................................... 10
      5.4.3 Purpose ........................................................ 10
      5.4.4 Function ....................................................... 11
      5.4.5 Sub-modules .................................................. 11
5.4.6 Dependencies ................................................................................. 11
5.5 Open response assessment ................................................................. 11
  5.5.1 Identification ............................................................................. 11
  5.5.2 Type ......................................................................................... 12
  5.5.3 Purpose .................................................................................... 12
  5.5.4 Function ................................................................................... 12
  5.5.5 Sub-modules ............................................................................. 12
  5.5.6 Dependencies ............................................................................ 13
5.6 Analytics .......................................................................................... 13
  5.6.1 Identification ............................................................................. 13
  5.6.2 Type ......................................................................................... 13
  5.6.3 Purpose .................................................................................... 14
  5.6.4 Function ................................................................................... 14
  5.6.5 Sub-modules ............................................................................. 14
  5.6.6 Dependencies ............................................................................ 15
5.7 Ed-insights(analytics framework) ...................................................... 15
  5.7.1 Identification ............................................................................. 15
  5.7.2 Type ......................................................................................... 15
  5.7.3 Purpose .................................................................................... 15
  5.7.4 Function ................................................................................... 16
  5.7.5 Sub-modules ............................................................................. 16
  5.7.6 Dependencies ............................................................................ 17

6. Design Diagrams ................................................................................ 17
  6.1 Use case diagram for system ......................................................... 17
  6.2 Activity diagram ............................................................................ 19
  6.3 Sequence diagram .......................................................................... 23
  6.4 Dataflow diagram .......................................................................... 30

7. Database Design ................................................................................ 36
  7.1 Introduction ................................................................................... 36
  7.2 SQLite .......................................................................................... 36
  7.3 Internal data formats in edX ........................................................... 41
  7.4 MongoDB ..................................................................................... 44
1. Introduction

1.1 Purpose:
This document provides a high level description of the design and implementation of the proposed project named as “iitbombayX” based on edX. “iitbombayX” is a platform for massive open online courses (MOOC). This Design Report show how the proposed software system will be structured in order to satisfy the requirements identified in the Software Requirements Specifications document. This document has been written for software engineers, software architecture and technical program managers.

1.2 Scope:
The System named as “iitbombayX” is basically a MOOC based on “edX”, where “edX” is itself a MOOC platform developed in Django framework. This system aims to provide an online education platform which can support a large traffic with a number of advanced features for both author and student to enhance the online learning and making it more effective. This project also aims to generate various types of analytics so that the system can be tailored according to the requirements, that will help the system to be adaptive and more user oriented in its design.
2. Definitions, Acronyms, Abbreviations

**MOOC**: It stands for Massive open online course, which is an online course aimed at large-scale interactive participation and open access via the web for education purpose.

**Instructor/Author**: It stands for the person who creates the course using studio (CMS).

**Student**: It stands for the person who get enrolls in the course and gives test submit assignment.

**Peer**: It stands for other students who also enroll in the same course as student and act as a reviewer for assignment submitted by particular student.

**System**: It refers to the system being developed which is “iitbombayX”, built on “edX”.

**SRS**: Software requirement specification.

3. References:

The following list consists of all references which is being taken for preparing this document:

3. [https://github.com/edx/](https://github.com/edx/)
4. Consideration for producing SDD

4.1 Software Life Cycle:
The project development model which applies in this project is known as “Agile Software development model”. This model includes a group of software development methods based on iterative and incremental development. Using this model allows the requirements and solution to evolve through the contribution of collaborative efforts of different teams. It allows evolutionary development, adaptive planning which encourages rapid and flexible response to change.

4.2 Purpose of a SDD:
The SDD serve the purpose of making design decision of the software which is going to be developed. The SDD presented here consist of the main functions and modules present in the system with their input and output (in all possible case). Also with their interaction is shown using various UML diagrams like sequence diagram, activity diagram etc. The SDD provides a blueprint of the structure of the actual system to be built and help to take decision during the implementation phase. It may also include some constraints that required by the system (“iitbombayX”) being developed.

5. Design Description information content:

5.1 Introduction:

The entire system is divided into 3 components:

- LMS: The learning management system where the students register themselves and interact with the system and peruse online education.
- CMS: The Course management system is the system component where the instructor will create various components of the course.
- Edge: Edge provides a platform similar to LMS to the instructor so that the instructor can preview the course created by them.

The overall structure of the system is modular. There are various modules in the system which provides or implements various features in the system.
5.2 edX-platform (Main LMS and CMS):

5.2.1 Identification:
The module is the main module named as “edX-platform”

5.2.2 Type:
The type of this component is “Module”

5.2.3 Purpose:
This module is being made for structuring the basic LMS and CMS. This is the main module which directly interacts with the system. All other module will interact to this system through integration to integrate all features to the basic LMS and CMS.

5.2.4 Function:
The function of this module is to interact with user through LMS and CMS.

5.2.5 Sub-modules:
The edX-platform is one of the biggest modules of the system which in itself consist of various other sub modules contributing towards whole developed system. They are:

- **LMS**: The main function of this module is to provide the templates for all interface, consist of various plugging for various course components like circuit. Define functions for data transfers between various other component of system. This mainly consist of all function oriented to the Learning management system to be used be the user.
• **CMS:** The main function of this module is to provide templates for all interfaces for CMS. Define functions and components to create course, upload video, files, transfer data etc. This module is being used only by the instructor of the course.

• **Other modules:** Apart from the two main modules mentioned above there are other modules with assist the main modules and also involves files for installation in server, settings etc.

5.2.6 Dependencies:
This module is the main module of whole system, it interacts with all other system modules. It passes log files to the analytics server. It passes assignment to Xqueue for assessment.

5.3 codejail:

5.3.1 Identification:
The module is the main module named as “codejail”

5.3.2 Type:
The type of this component is “Module”.

5.3.3 Purpose:
The purpose of this module is to provide security to the system server so that it can survive the execution of the harmful code. This module is basically used for assessments of the submitted assignment which involves execution of codes.
5.3.4 Function:
The function of this module is to provide sandbox for secure execution of untrusted code.

5.3.5 Sub-modules:
The Code jail consist of only one submodule and other files for settings etc.
- **codejail**: This is the only module present in the in code jail which consist of the main files defining function for creating a virtual environment, to configure a command for execution under code jail, to run commands of codes in code jail etc.

5.3.6 Dependencies:
This module takes python code for secure execution from the LMS and the output is then returned to LMS.

5.4 Discussion forum:

5.4.1 Identification:
The module is the main module named as “cs_comments_service”.

5.4.2 Type:
The type of this component is “Module”

5.4.3 Purpose:
The purpose of this module is to support discussions which involves voting, nested comment in both LMS and CMS/
5.4.4 **Function:**
The main function of this module is to handle the discussion forum activities.

5.4.5 **Sub-modules:**
It consist of few submodules to divide the work of the main module. The sub modules present in cs_comment service are:

- **api:** The main function of this module is to provide various api to associate user id with the discussion post or comment, to read, search discussion etc.

- **Other modules:** Apart from the two main modules mentioned above there are other modules with assist the main modules and also involves files for installation in server, settings, have helper functions, Rake functions, functions to make user anonymous identities for peers etc.

5.4.6 **Dependencies:**
This module interacts with edX-plaform (LMS and CMS). The discussion post and comments are passed from the LMS and CMS to cs_comment_service there various functions handle these post pass them to store and also to extract and pass them to LMS and CMS.

5.5 **Open response assessment:**

5.5.1 **Identification:**
The module is identified in the system by the name “edX-ORA”.

5.5.2 **Type:**
The type of this component is “Module”
5.5.3 **Purpose:**
The purpose of this module is to provide functions to assess the submitted assignment. It consists of programs to take assignment from the LMS, submitted by the user and assess the assignment and return grades back to the LMS (also storing it in database).

5.5.4 **Function:**
The main function of this module is to provide the functions for assessments of submitted essay. It provides four types of assessment for essay and calculates the grades and returns it to the LMS.

5.5.5 **Sub-modules:**
It consists of few submodules to deal with different types of assessment being provided by the system accompanied with other modules to provide templates and helper functions etc. The main modules are listed as follows:

- **Peer_grading:** This module provides functions for performing various steps of peer grading for example it includes function for calibration round, The functions to decide the eligibility of user to grade essay, functions to extract essay using user_id etc.

- **Staff_grading:** This module provides functions for performing various steps for staff grading of the essay by the staff. for example it includes function for authentication of staff member, functions for return essay to corresponding staff member, etc.
- **Controller:** This module provides functions for controlling various grading being performed by the system also it includes functions to parse rubric objects, items, checking for expiration of the submission, reset expiration times, check for number of essay pending to grade etc.

- **ML_grading:** This module deals with the data models related to the grading, it deals with taking data from one LMS server and providing data to other module’s function.

- **Other modules:** Other modules present in the edX-ORA takes care of providing templates, static files, configuration and settings etc.

5.5.6 **Dependencies:**
This module is the main module of whole system, it interacts with all other system modules takes assignments through xqueue (another module present in edX) and pushes back the graded assessment to the xqueue.

5.6 **Analytics:**

5.6.1 **Identification:**
The component is name as “edxanalytics”.

5.6.2 **Type:**
The type of this component is “Module”

5.6.3 **Purpose:**
The purpose of this module is to generate various kinds of analytics based on the events occurred while using the LMS and CMS by the users to provide various kinds of statistics so that the instructor or the system admin can look at the statistics of the usage of the system and tailor it accordingly to make system more effective.

5.6.4 Function:
The main function of this module is to process the log files, extract various parameters from the files, formulate the analytics and generate user dashboards accordingly.

5.6.5 Sub-modules:
It consists of few submodules to achieve main motto of processing the log files to generate proper analytics. The main modules present in it is enlisted below:

- **src**: This is the main module of the edxanalytics which might consist of some submodules itself. This module consists of all static files required to create the user dashboard. Other sub modules under it is described as follows:
  - edx_embed: This module consists of the function to embed the analytics in the LMS server.
  - Edxanalytics: This module consists of the functions to perform analytics on the various log files and to handle django events. It consists of functions to extract data from the log events and calculate some parameters and generate analytics on that.
• **Edxdataanalytics:** This module consists of functions to exchange data with ed-insights and export data to main analytics server.

• **Other module:** other modules may involve which might consist of setting files, files for deployment of the edxanalytics in the server, templates and utility functions.

5.6.6 **Dependencies:**
This module interacts with LMS indirectly through djeventstream and loghandlersplus for log files and django event signals. It also interacts with ed-insights for using the functions and views defined in it.

5.7 **Ed-insights(analytics framework):**

5.7.1 **Identification:**
The component is identified as “Ed-insights”.

5.7.2 **Type:**
The type of this component is “framework module”

5.7.3 **Purpose:**
The purpose of this module is to act as a framework to create pluggables analytics based on streaming events. It also involves basic functions for processing log files and creating analytics based on that.
5.7.4 **Function:**
The main function of this module is to define analytics based on the framework involving django+python+mongodb.

5.7.5 **Sub-modules:**
It consist of few submodules to divide the various parts of framework efficiently. They are listed as below:

- **src:** This is the main module of the ed-insights which consist of all function to which acts as the framework for the development analytics. The types of functions defined in this modules are:
  1. Decorator functions.
  2. Helper functions
  3. Views to design dashboard of user.
  4. Functions to authenticate user of analytics.
  5. Functions to generate
  6. Handles events (as queue or batch)
  7. Etc.

- **Examples:** These modules consist of various functions to perform few kinds of analytics which is performed by default by the system. For example: functions are to be defined video heat maps which perform analytics on videos being watched by the user.

- **Other module:** Other modules may involve setting files, files for deployment of the ed-insights in the server, templates and utility functions.
5.7.6 Dependencies:
This module interacts with LMS by taking assignments as input and provides this assignment to external grader which is “edX-ORA” and also takes the graded assignment back from ORA and provide results to LMS.

6. Design Diagrams

6.1 Use case diagram for system
This diagram shows the interaction of the users with the system being developed.
6.2 Activity diagram:

This diagram is basically a workflow of the stepwise activities and various actions presented along with the choice, repetitions provided by the system.

Activity diagram for student:
Activity diagram for instructor in LMS:
Activity Diagram for self and peer evaluation of assignment:

1. Self evaluation
2. View Score
3. Grade Essay for calibration
   - If satisfied: If not satisfied: Reload assignment
   - If not satisfied: Grade Essay for peer
     - If satisfied: Submit Grades
     - If not satisfied: passed calibration
       - If not satisfied: Reload assignment
       - If satisfied: Grade Essay for peer
         - If satisfied: Submit Grades
         - If not satisfied: Reload assignment
Activity diagram for Instructor in CMS:
6.3 Sequence Diagram:

This diagram is made to understand the interaction between different objects of the system also their sequence of interaction with each other.

**Sequence Diagram for registration**

User

- Request for registration
  - Provide register interface
    - Submit user details
      - Prompt for new user/mail id
        - final user details
          - User dashboard
            - Activation mail
              - user confirms registration

System

- Check for valid and non-empty input
  - Check for unique user/mail id
    - If not unique
      - Forward user details to save
        - Registration successful
          - Activate user account

Database
Sequence diagram for Login of user (both student and author)
Sequence diagram for performing search for course:

1. User (represented as a stick figure) enters a search keyword.
2. The system receives the search keyword and forwards it to the database.
3. The database performs the search.
4. The search result is returned to the system.
5. The system formulates the proper view and returns the result of the search to the user.
Sequence diagram for Discussion forum (both student and Instructor):
Sequence Diagram to upload file by user (both student and instructor)
Sequence diagram to review assignments:

1. Student
   - Prompt for Review Assignments
   - Opt to review
   - Display List
   - Select 1 student
   - Display feedback form with assignment
   - Submit grades/feedback
   - Success Message

2. System
   - Forward
   - Send List of peers (whose Assignment is to be graded)
   - Request to fetch assignment
   - Result of request
   - Validates
   - Save Grades and feedback
   - Success

3. Database
   - Failure
   - Failure Message
Sequence diagram to add new course (by instructor):

- Instructor
  - Request to add course
    - Provides form
    - Fills form
    - Prompts for error
    - Validate submission
    - Forward details
    - Failure if Id already exist
    - Check for unique course id
      - Saves course info
    - Success
    - Saves course info
  - Give failure message
    - If continue
      - Gives success message
      - Request to add new section
        - Provides interface
        - Fills form
        - Provides course interface
        - Request to add new subsection
          - Provides interface
          - Submit subsection info
        - Success message
        - Saves section info
      - Saves section info
  - Request to add new unit
    - Provides interface
    - Selects unit type
    - Provide unit specific interface
    - Fills form/edit content
    - Validate submission
    - Forward details
    - Success
  - Success message
    - Saves section info
6.4 Dataflow Diagram:
The data flow diagram presented here is to show the flow of data in the system between different components of the system.

Context Diagram:
Level 0 diagram:

1. Registration
   - Registers by providing valid details
   - Logs in with his registered email id and password
   - Manages his/her profile
   - Register for a course and access course content

2. Login
   - Logs in with his registered email id and password
   - Manages his/her profile
   - Search for a course

3. Profile Maintenance
   - Registers by providing valid details
   - Logs in with his registered email id and password
   - Manages his/her profile

4. Course Management
   - Registers by providing valid details
   - Logs in with his registered email id and password
   - Manages his/her profile
   - Add/delete course on the site and add/release date for the course

5. Site Maintenance
   - Manages site (logo, UI, etc.)
   - Add/edit articles
   - Correct submit assignments
   - Create exams/test questions and create rubrics
   - Can see progress report

6. Searching
   - Manages site (logo, UI, etc.)
   - Add/edit articles
   - Correct submit assignments
   - Create exams/test questions and create rubrics
   - Can see progress report

7. Peer Assessment
   - Contribute to discussions and ask questions
   - Add/edit articles
   - Correct submit assignments
   - Create exams/test questions and create rubrics
   - Can see progress report

8. Group Discussion
   - Contribute to discussions and ask questions
   - Add/edit articles
   - Correct submit assignments
   - Create exams/test questions and create rubrics
   - Can see progress report

9. Wikileaks
   - Can see progress report

10. Assignment Submission
   - Can see progress report

11. Test Exam
   - Can see progress report

12. Progress Report
   - Can see progress report
Level 1 diagram:

Register dfd:

Login dfd:
Discussion dfd:

Course management dfd:
Profile maintenance dfd:

Wiki edit dfd:
Report dfd:

![DFD Diagram]

- **Student**
  - Grade returned

- **Progress Report**
  - Visible to the user after grading
  - Evaluates assignments

- **Instructor**
  - Grade added/updated

- **courseware_offlinecomputedgraded**
7. Database Design:

7.1 Introduction:
The major requirements for database in edX is for storing the user information details, courseware information, student module information, edX wiki information, course content, metadata etc. edX is using two databases in parallel to manage content on the data one is Sqlite which is the relational database and MongoDB which is a NoSQL non-relational database.

7.2 Sqlite:
This is a relational database system where various tables are formed in order to store data in a conventional way which includes storing of tuples having various attributes. For local development environment Sqlite has been used in order to keep the dependencies simple but for production environments mysql comes into role. As edX is majorily built on the django framework thus all the tables in the django's legacy database are also included in the edX database.

In the edX platform the role of Sqlite is to store the user profile, user information, course and student modules, courseware details, sample wiki data, enrollment data, licensing, certification, test center and registration details. When a course is created it creates two groups for staff and instructor in the and also defines the permission roles for the users of that course.

In the django's legacy database, initially there are 9 tables which include user, user_groups , groups, permissions, user and group permissions, django site ,session and django content type. So the django's database ER diagram looks like:
and the details of the relations in the django's legacy database are:
After installing the edX database the data is synchronized first, creates the tables, migrates the indexes and stores them in mitx.db which can be viewed through sqlite3 interface. In this relational format there are 85 tables created which include in it also the tables of the django's legacy database. These tables store mainly the account details, courseware information, student module information, etc.. The basic information of users are included in auth_user and auth_userprofile ofile. The example of schema(of auth_user and auth_userprofile respectively) are:
CREATE TABLE "auth_user" ("username" varchar(30) NOT NULL UNIQUE, "first_name" varchar(30) NOT NULL, "last_name" varchar(30) NOT NULL, "is_active" bool NOT NULL, "id" integer PRIMARY KEY, "is_superuser" bool NOT NULL, "is_staff" bool NOT NULL, "last_login" datetime NOT NULL, "password" varchar(128) NOT NULL, "email" varchar(75) NOT NULL UNIQUE, "date_joined" datetime NOT NULL);

CREATE TABLE "auth_user" ("username" varchar(30) NOT NULL UNIQUE, "first_name" varchar(30) NOT NULL, "last_name" varchar(30) NOT NULL, "is_active" bool NOT NULL, "id" integer PRIMARY KEY, "is_superuser" bool NOT NULL, "is_staff" bool NOT NULL, "last_login" datetime NOT NULL, "password" varchar(128) NOT NULL, "email" varchar(75) NOT NULL UNIQUE, "date_joined" datetime NOT NULL);

The classification of tables in this file is done in various parts depicted below:
edX database tables

Courseware Information
- courseware_offlinecomputedgrade
- courseware_offlinecomputedgradelog
- courseware_studentmodule
- courseware_studentmodulehistory
- courseware_xmodulecontentfield
- courseware_xmodulesettingsfield
- courseware_xmodulestudentinfofield
- courseware_xmodulestudentpreferences

User and Group Information
- auth_group
- auth_group_permissions
- auth_permission
- auth_registration
- auth_user
- auth_user_groups
- auth_user_user_permissions
- auth_userprofile

Session and tracking logs
- django_session
- django_site
- track_trackinglog

Student Course Enrollment and other details
- student_courseenrollment
- student_courseenrollment_moduleenrollmentallowed
- student_pendingemailchange
- student_pendinglearnchange
- student_testcenterregistration
- student_testcenteruser
- student_usersgroup
- student_usersgroup_users

edX Wiki
- wiki_article
- wiki_articleforobject
- wiki_articleplugin
- wiki_articleversion
- wiki_articlesubscription
- wiki_attachment
- wiki_attachmentversion
- wiki_image
- wiki_imagerevision
- wiki_reusableplugin
- wiki_reusableplugin_articles
- wiki_revision
- wiki_revisionplugin
- wiki_revisionpluginversion
- wiki_simpleplugin
- wiki_unipath

NOTE:
- tables present in django legacy database

django comment service information
- django_admin_log
- django_comment_client_permission
- django_comment_client_permissions
- django_comment_client_role
- wiki_module
- django_comment_client_role_users
- django_content_type
- django_openid_auth_association
- django_openid_auth_nonce
- django_openid_auth_useropenid
These are the various categories in which the tables can be organized and the relation between these relations have been included in the ER diagram of the database has been included in the appendix of the project report.

### 7.3 Internal Data Formats in edX:

This has three major categories: Data regarding student information and progress, Data regarding log tracking and Data for Discussion forums.

Student Information - Progress Data: Contains user and profile information with the information of courses enrolled in. Schema of these relations are:

**User information:**

**auth_user:**

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Null</th>
<th>Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>int(11)</td>
<td>NO</td>
<td>PRI</td>
</tr>
<tr>
<td>username</td>
<td>varchar(30)</td>
<td>NO</td>
<td>UNI</td>
</tr>
<tr>
<td>first_name</td>
<td>varchar(30)</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>last_name</td>
<td>varchar(30)</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>email</td>
<td>varchar(75)</td>
<td>NO</td>
<td>UNI</td>
</tr>
<tr>
<td>password</td>
<td>varchar(128)</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>is_staff</td>
<td>tinyint(1)</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>is_active</td>
<td>tinyint(1)</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>is_superuser</td>
<td>tinyint(1)</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>last_login</td>
<td>datetime</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>date_joined</td>
<td>datetime</td>
<td>NO</td>
<td></td>
</tr>
</tbody>
</table>
**auth_userprofile**

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Null</th>
<th>Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>int(11)</td>
<td>NO</td>
<td>PRI</td>
</tr>
<tr>
<td>user_id</td>
<td>int(11)</td>
<td>NO</td>
<td>UNI</td>
</tr>
<tr>
<td>name</td>
<td>varchar(255)</td>
<td>NO</td>
<td>MUL</td>
</tr>
<tr>
<td>language</td>
<td>varchar(255)</td>
<td>NO</td>
<td>MUL</td>
</tr>
<tr>
<td>location</td>
<td>varchar(255)</td>
<td>NO</td>
<td>MUL</td>
</tr>
<tr>
<td>meta</td>
<td>longtext</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>courseware</td>
<td>varchar(255)</td>
<td>NO</td>
<td>MUL</td>
</tr>
<tr>
<td>gender</td>
<td>varchar(6)</td>
<td>YES</td>
<td>MUL</td>
</tr>
<tr>
<td>mailing_address</td>
<td>longtext</td>
<td>YES</td>
<td>**</td>
</tr>
<tr>
<td>year_of_birth</td>
<td>int(11)</td>
<td>YES</td>
<td>MUL</td>
</tr>
<tr>
<td>level_of_education</td>
<td>varchar(6)</td>
<td>YES</td>
<td>MUL</td>
</tr>
<tr>
<td>goals</td>
<td>longtext</td>
<td>YES</td>
<td>**</td>
</tr>
<tr>
<td>allow_certificate</td>
<td>tinyint(1)</td>
<td>NO</td>
<td></td>
</tr>
</tbody>
</table>

**Only users signed up after prototype**

**student_courseenrollment**

*Id*- Primary key

*user_id*- references auth_user.id.

*course_id*- id of course enrolled in.

*Created*- Date and time of enrollment

**courseware_studentmodule**

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Null</th>
<th>Key</th>
<th>[5]</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>int(11)</td>
<td>NO</td>
<td>PRI</td>
<td></td>
</tr>
<tr>
<td>module_type</td>
<td>varchar(32)</td>
<td>NO</td>
<td>MUL</td>
<td></td>
</tr>
<tr>
<td>module_id</td>
<td>varchar(255)</td>
<td>NO</td>
<td>MUL</td>
<td></td>
</tr>
<tr>
<td>student_id</td>
<td>int(11)</td>
<td>NO</td>
<td>MUL</td>
<td></td>
</tr>
<tr>
<td>state</td>
<td>longtext</td>
<td>YES</td>
<td></td>
<td></td>
</tr>
<tr>
<td>grade</td>
<td>double</td>
<td>YES</td>
<td>MUL</td>
<td>**</td>
</tr>
<tr>
<td>created</td>
<td>datetime</td>
<td>YES</td>
<td>MUL</td>
<td></td>
</tr>
<tr>
<td>modified</td>
<td>datetime</td>
<td>NO</td>
<td>MUL</td>
<td></td>
</tr>
<tr>
<td>max_grade</td>
<td>double</td>
<td>YES</td>
<td></td>
<td>**</td>
</tr>
<tr>
<td>course_id</td>
<td>varchar(255)</td>
<td>NO</td>
<td>MUL</td>
<td></td>
</tr>
</tbody>
</table>

[5]
Tracking log information is included in the track_tackinglog, for every action in the front end there is a tuple created in this relation.

This covers the structure and organization of tables in the relational database, but for the course content storage the document based non relational database – Mongodb is used.

The discussion forum data is stored in Mongodb as comment and comment threads covered later.

**Course Data Formats:**
Course data is stored in xml format which includes course content and course overview:
Example of .xml file:
// main course.xml file contains the description
<course org="iitbX" course="DSA" url_name="2013_Spring"/>
//Course content
<!-- Contents of course/2013_Spring.xml-->
<course>
  <chapter url_name="sem_details">
    <videosequence url_name="sem_videos">
      <problem url_name="ece"/>
      <video url_name="Video_Resources" youtube="2.0:2gK-CdDe3Qw"/>
    </videosequence>
    <video url_name="HelloWorld" youtube="2.0:t8R6BrVhdh8"/>
  </chapter>
</course>

Container tags are there in xml files for course information. Available container tags are cursor, custom, discussion, videos, problem, custom etc.

And the other information about policies, location of tabs, pdfs, and textbooks are stored in .json formats.

7.4 MongoDB

Since Mongodb is a non-relational NoSql document based database thus facilitates to organize and store the course content. It has no predefined schema rather uses a dynamic schema in form of collections (which contains documents) in JSON formats. Collection are different form table as, they don't have fixed and predefined typed set of attributes which are there in relational databases. The Documents are different from tuples as, they don't need to give information for all the attributes.
Each document contains _id information which uniquely identifies a document serves as a primary key.

Key-value pairs are created for the data stored in Mongodb databases so it can be optimized whenever any kind of operation is performed in these databases. Thus Mongodb is much scalable and faster.

Using MongoDB as a content store new and provides flexibility in managing the Json documents.

In edX there are two databases of MongoDB format: xmodule which stores module definitions and metadata and xcontent which stores file system content.

Example of a 'problem' in MongoDB document

```json
{
    "_id" : {
        "tag" : "i4x",
        "org" : "IITBx",
        "course" : "CSE1x",
        "category" : "problem",
        "name" : "Stacks_Queue2",
        "revision" : null
    },
    "definition" : {
        "data" : "...

    },
    "metadata" : {
```
"display_name" : "Stacks_QUEUES2",
"attempts" : "30",
"title" : "Part 2",
"data_dir" : "CSE1x",
"type" : "lecture"

Similarly Discussion forum data are also stored in the json scripts which contains data organized comments and comment thread and has some shared attributes like course_id, create date, votes etc.