

**PROPOSAL:**

**Curriculum Management System**

**Name 1**

**Submitted by**

Mohi Karim

Verbanet Technologies LLC  
12.03.19

**DIRECTORY.**

[**1** **KEY DETAILS** 8](#_Toc3993446)

[**2** **STRATERGIC OBJECTIVES** 9](#_Toc3993447)

[**3** **PROJECT SCOPE** 10](#_Toc3993448)

[3.1 DEFENITION of TERMS 11](#_Toc3993449)

[3.2 USER CHARACTERISTICS 13](#_Toc3993450)

[3.2.1 Generic user 13](#_Toc3993451)

[3.2.2 Faculty/staff user 14](#_Toc3993452)

[3.2.3 Admin user 14](#_Toc3993453)

[3.2.4 Workflow User: Can also be in one of the three previous categories 14](#_Toc3993454)

[3.2.5 Sample Create Course Workflow, Roles, Permissions 14](#_Toc3993455)

[3.3 PROPOSED SOLUTION MODEL 16](#_Toc3993456)

[3.4 FUNCTIONAL REQUIREMENTS 17](#_Toc3993457)

[3.5 COURSE MANAGEMENT 18](#_Toc3993458)

[3.5.1 Course Information 18](#_Toc3993459)

[3.5.2 Course Processes 20](#_Toc3993460)

[3.6 PROGRAM MANAGEMENT 21](#_Toc3993461)

[3.6.1 Program Information 21](#_Toc3993462)

[3.6.2 Program Processes 22](#_Toc3993463)

[3.7 COURSE SET MANAGEMENT 23](#_Toc3993464)

[3.8 LEARNING OBJECTIVES MANAGEMENT 23](#_Toc3993465)

[3.9 ORAGANIZATION, PEOPLE and ROLES MANANAGEMENT 24](#_Toc3993466)

[3.10 WORKFLOW STATES 26](#_Toc3993467)

[3.11 PROPOSAL STATES 26](#_Toc3993468)

[3.12 COURSE STATES 27](#_Toc3993469)

[3.13 COURSES and PROPOSAL WORKFLOW with STATES 29](#_Toc3993470)

[4 APPLICATION SERVICES 30](#_Toc3993471)

[4.1 Academic Time Period Service 30](#_Toc3993472)

[4.1.1 Description 30](#_Toc3993473)

[4.1.2 Key Concepts 30](#_Toc3993474)

[4.2 Canonical Learning Unit (CLU) Service 30](#_Toc3993475)

[4.2.1 Description 30](#_Toc3993476)

[4.2.2 Key Concepts 31](#_Toc3993477)

[4.3 Comment Service 33](#_Toc3993478)

[4.3.1 Description 33](#_Toc3993479)

[4.3.2 Key Concepts 34](#_Toc3993480)

[4.4 Course Service 34](#_Toc3993481)

[4.4.1 Description 34](#_Toc3993482)

[4.4.2 Key Concepts 35](#_Toc3993483)

[4.5 Document Service 36](#_Toc3993484)

[4.5.1 Description 36](#_Toc3993485)

[4.5.2 Key Concepts 36](#_Toc3993486)

[4.6 Enumeration Management Service 37](#_Toc3993487)

[4.6.1 Description 37](#_Toc3993488)

[4.6.2 Key Concepts 37](#_Toc3993489)

[4.7 Learning Objective Service 37](#_Toc3993490)

[4.7.1 Description 37](#_Toc3993491)

[4.7.2 Key Concepts 37](#_Toc3993492)

[4.8 Learning Result Catalog Service 38](#_Toc3993493)

[4.8.1 Description 38](#_Toc3993494)

[4.8.2 Key Concepts 38](#_Toc3993495)

[4.9 Message Service 39](#_Toc3993496)

[4.9.1 Description 39](#_Toc3993497)

[4.9.2 Key Concepts 40](#_Toc3993498)

[4.10 Organization Service 40](#_Toc3993499)

[4.10.1 Description 40](#_Toc3993500)

[4.10.2 Key Concepts 40](#_Toc3993501)

[4.11 Program Service 40](#_Toc3993502)

[4.11.1 Description 40](#_Toc3993503)

[4.11.2 Key Concepts 41](#_Toc3993504)

[4.12 Proposal Service 42](#_Toc3993505)

[4.12.1 Description 42](#_Toc3993506)

[4.12.2 Key Concepts 42](#_Toc3993507)

[4.13 Search Service 43](#_Toc3993508)

[4.13.1 Description 43](#_Toc3993509)

[4.13.2 Key Concepts 44](#_Toc3993510)

[4.14 State Service 44](#_Toc3993511)

[4.14.1 Description 44](#_Toc3993512)

[4.14.2 Key Concepts 44](#_Toc3993513)

[4.15 Version Management Service 45](#_Toc3993514)

[4.15.1 Description 45](#_Toc3993515)

[4.15.2 Key Concepts 46](#_Toc3993516)

[5 DETAILED REQUIRMENT SPECIFICATIONS 47](#_Toc3993517)

[6 APPLICATION ARCHITECHURE 54](#_Toc3993518)

[6.1 COURSE CREATION WORKFLOW 54](#_Toc3993519)

[6.2 ENTITY RELATIONSHIP DIAGRAMS 55](#_Toc3993520)

[6.2.1 ATP Service 55](#_Toc3993521)

[6.2.2 Canonical Learning Unit 56](#_Toc3993522)

[6.2.3 Comment Service 57](#_Toc3993523)

[6.2.4 Course Service 58](#_Toc3993524)

[6.2.5 Document Service 59](#_Toc3993525)

[6.2.6 Enumeration Management Service 60](#_Toc3993526)

[6.2.7 Learning Objective Service 61](#_Toc3993527)

[6.2.8 Learning Result Catalog Service 62](#_Toc3993528)

[6.2.9 Message Service 63](#_Toc3993529)

[6.2.10 Organization Service 64](#_Toc3993530)

[6.2.11 Program Service 65](#_Toc3993531)

[6.2.12 Proposal Service 66](#_Toc3993532)

[6.2.13 State Service 67](#_Toc3993533)

[6.2.14 Version Management Service 67](#_Toc3993534)

[6.3 LOCAL SERVER INSTANCE (On Premise) 68](#_Toc3993535)

[6.4 Sample Screens 69](#_Toc3993536)

[6.4.1 Course List 69](#_Toc3993537)

[6.4.2 Course View 70](#_Toc3993538)

[6.4.3 Form Changes (Tracking) 71](#_Toc3993539)

[6.4.4 Rich Text Content 72](#_Toc3993540)

[6.4.5 Tagging Courses 73](#_Toc3993541)

[6.5 NON-FUNCTIONAL REQUIREMENTS (OTHERS) 74](#_Toc3993542)

[6.6 TECHNICAL CONFIGURATIONS 74](#_Toc3993543)

[6.6.1 DEVELOPMENT ENVIRONMENT 74](#_Toc3993544)

[6.6.2 RECOMMENDED WED HOSTING PACKAGE - DEDICATED 75](#_Toc3993545)

[6.6.3 BROWSER 75](#_Toc3993546)

[6.6.4 HARDWARE DEVICES 75](#_Toc3993547)

[6.6.5 TECHNICAL STANDARDS 76](#_Toc3993548)

[6.6.6 TECHNICAL GUIDELINES 76](#_Toc3993549)

[6.7 PROJECT DELIVERY 78](#_Toc3993550)

[6.7.1 Agile Methodology 78](#_Toc3993551)

[6.7.2 PROJECT MANAGEMENT 80](#_Toc3993552)

[6.7.3 ROLES & RESPONSIBILITIES 81](#_Toc3993553)

[6.7.4 DELIVERY ACTIVITY SUMMARY 82](#_Toc3993554)

[6.7.5 PROJECT IMPLEMENTATION PLAN 83](#_Toc3993555)

[6.7.6 DELIVERABLES 83](#_Toc3993556)

[6.7.7 ESTIMATED DELIVERY TIME 84](#_Toc3993557)

[6.7.8 DEPLOYMENT DETAILS 86](#_Toc3993558)

[6.7.9 RELEASE PLANNING 86](#_Toc3993559)

[6.7.10 RISK CONTINGENCY PLANNING 86](#_Toc3993560)

[6.8 PROJECT ASSUMPTIONS 88](#_Toc3993561)

[6.8.1 OBJECTIVES 88](#_Toc3993562)

[6.8.2 DESIGN 88](#_Toc3993563)

[6.8.3 DEVELOPMENT 88](#_Toc3993564)

[**7** **OUT OF SCOPE** 89](#_Toc3993565)

[**8** **CHANGE MANAGEMENT** 91](#_Toc3993566)

[8.1 MAINTENANCE & SUPPORT 92](#_Toc3993567)

[**9** **TERMS AND CONDITIONS** 93](#_Toc3993568)

[9.1 ACCEPTANCE CRITERIA 93](#_Toc3993569)

[9.2 WARRANTY 93](#_Toc3993570)

[9.3 SOURCE CODE & INTELLECTUAL PROPERTY RIGHTS 94](#_Toc3993571)

[9.4 GENERAL TERMS AND CONDITIONS 94](#_Toc3993572)

[9.5 GENERAL ADMINISTRATIVE, TECHNICAL & FUNCTIONAL ASSUMPTIONS 96](#_Toc3993573)

[**10** **FINANCIALS** 97](#_Toc3993574)

[10.1 Curriculum Management Application 97](#_Toc3993575)

[10.2 PAYMENT TERMS 97](#_Toc3993576)

[10.2.1 MODE OF PAYMENT 98](#_Toc3993577)

# **KEY DETAILS**

**PROJECT NAME CLIENT**

Curriculum Management System Client 1

**CLIENT CONTACT PROPOSAL ID NO.**

Name 1 AD/BP/12032019/2609/1

m.aldarabee@gmail.com

+966 59 011 1298

**PROPOSAL SUBMISSION ANTICIPATED START DATE**

12.03.19  TBD

**PROPOSAL VALID UNTIL PROPOSED TECHNOLOGY**

11.04.19 JAVA, SPRING

CSS3, HTML5

Responsive Web Application

**PROPOSAL SUBMITTED BY APPLICATION TYPE**

Mohi Karim Web & Mobile Application

Verbanet Technologies L.L.C

Dubai, U.A.E

+971 4 297 3236

Mohi.karim@verbat.com

# **STRATERGIC OBJECTIVES**

MOE has requested for proposals to develop a Curriculum Management System for the public educational system (Pre-K up to grade 12 with potential to integrate with Higher Education). The Curriculum Management System encompasses the whole spectrum of activities related to the development of the curriculum framework. The framework shall be composed of

* Course Management
  + Course Information
  + Course Processes
* Program Management
  + Program Information
  + Program Processes
* Course Set Management
* Learning Objectives Management
* Organization, People and Roles Management
* Course and Proposal States
* Reference Implementation and Configuration
  + Organization configuration
  + Subject Areas and Curriculum Org Relationship
  + User Configuration
  + Workflow Configuration and Associated Users
* Catalog Management
* Version Management
* Archiving
* Analytics and Reports
* Integration with External API’S
* SSO

# **PROJECT SCOPE**

Verbanet Technologies L.L.C., (hereafter referred under its trade / brand name as” Verbat”) in partnership with MOE (the Client) shall develop a Curriculum Management System. The system shall be a Bi-lingual (Arabic & English) Responsive Web Application. The application being developed shall have the following features

* Automating the process of developing, managing & modifying standards in the curricular framework
* Documentation of the accreditation process through a pre-defined workflow
* Automating the approval process of the school matrix and study plan in a systematic manner
* Automatically archiving the different versions of the standards – on a periodic manner – with all other curriculum components (content, resources, tools & policies) related to them
* Archiving all curricula components per academic year and provide full access to them via a searching tool
* Ensuring a historical audit trail of any changes with comprehensive reports that show changes made, changes approved, date of the change and approval of the persons who made them
* Providing access to all sectors in the ministry and ensuring their engagements in the review and approval process
* Ensuring flexibility in user management; types of users available; their roles and scopes
* Creating catalogues for each subject, grade level and stream that can be shared with different stakeholders (students, teachers, universities, principals etc.)
* Developing and managing the assessment policy for each subject, grade level per educational system
* Creating follow-up reports that track all changes in the curricula and share them with a specific list of stakeholders
* Linking the public education system with the higher education system by demonstrating shared learning standards in the two systems providing mapping tools and reports to carry out the practice
* Developing different type of reports demonstrating horizontal and vertical alignment in the learning standards among the different subjects and classes
* Linking the human resources needs of teachers to the system
* Providing accounts to private schools in order to formally share their curricula with the Ministry of Education
* Facilitating the equivalency and accreditation process for different educational systems in the UAE through equivalence credit systems
* Linking the approved systems with EmSAT results
* Training MoE team on the optimal use of CMS and provide support on 24/7 basis
* Provide search feature that facilitates easy access of content to a pre-defined set of stakeholders; curriculum developers, administrators and moderators
* Enabling tagging at the level of learning outcomes
* Providing an Arabic supported search engine that is sensitive to diacritical marks
* Providing flexibility for the different data structure as per subject types and needs
* Supporting Arabic text in all levels (content and user interface)
* Customizable lookups
* Report extraction – PDF, Excel, HTML
* Solution should be hosted on premise
* SSO integration based on OpenConnect ID: Ability to creates users and roles locally
* API integration with other systems
* Monitoring tools to monitor the system

## DEFENITION of TERMS

|  |  |
| --- | --- |
| Term | Definition |
| Catalog | The collection of approved learning experiences that an institution may offer to its students. |
| Academic Program | aka, Major; An organized curriculum offered as a major area of study recognized as an academic discipline by the higher education community as part of a credential program |
| Certificate | An organized curriculum leading to a certificate in an area of study. |
| Core Program | A prescribed grouping of courses that is required for all students as a foundation of general knowledge. Also known as General Education |
| Course | A learning experience that imparts education through a series of activities such as lectures, labs, recitations, etc. within a well-defined time period; it results in an assessment (e.g., grade) and an outcome (e.g., credits). |
| Course Set | A set composed of one or more Courses |
| Credential Program | A prescribed grouping of learning experiences that leads to a credential. |
| Credit Course | A course which, upon successful completion, results in the award of academic credit. |
| Dynamic Course Set | A Course Set whose elements are defined implicitly by some criteria, e.g. (100-level math courses); were a new 100-level MATH course be added to the curriculum, it would automatically be added to this course set as well. |
| Experiential Learning | An individualized student learning experience that emphasizes learning by "doing" rather than "producing." Examples include service learning, internships, clerkships, and co-operative education. |
| Fixed Course Set | A Course Set whose elements are defined explicitly by the course number, e.g. (MATH101, MATH102, MATH103) |
| Governed Learning Objectives | Learning Objectives that are defined by a group of people external (e.g. accreditation bodies) or internal (e.g. a department, or an organization of biology, chemistry, and physics departments) to the institution. These Learning Objectives may only be modified by the internal organization of people who entered them with the expectation that these centrally managed, 'governed' learning objectives may be imported directly into courses or programs. |
| Learning Objective | A statement, usually written by a faculty member, that describes the knowledge, skills, or abilities that a student would expect to gain by participating in a learning experience |
| Non-Credit Course | A course which, upon successful completion, results in the award of non-academic units, such as Continuing Education Units. |
| Programs | A prescribed grouping of learning experiences that leads to a recognized body of knowledge; the end result of a completed program may be an acknowledged level of accomplishment (e.g. a major, minor or concentration) or a credential (e.g. certificate, degree, etc.) |
| Project | An individualized student learning experience designed to culminate in a "product” that synthesizes the learning obtained by the student. Examples of projects include theses, dissertations, research projects, and/or capstone projects. An important attribute of the project is the supervisory committee and associated rules regarding its make-up. Typically, projects will be associated with degree and certificate programs. |
| Proposal | A description of a learning experience, typically authored by faculty, that is submitted for review by the appropriate curricular oversight organization; if approved, the learning experience becomes part of the institution's curriculum |
| Reusable Course Set | A reusable Course Set is one which is created prior to building a rule, named, and saved, and is available for use in other rules. |
| Single-Use Course Set | A single-use Course Set is one which is created 'on the fly' when building a rule and is not available for use in other rules |
| Single-Use Learning Objectives | Learning Objectives that are defined within the context of a specific Course or Program and are not maintained as part of a centralized inventory. |
| Specializations | Variations and/or refinements of an Academic Program that lead to specific educational or occupational goals. |
| Tests | A procedure for evaluating learning of student; may be used to award credit and/or fulfill a requirement. |
| MOE | Ministry of Education |
| CLU | Canonical Learning Unit |
| ATP | Academic Time Period |
| LO | Learning Objective |

## USER CHARACTERISTICS

### Generic user

* Can generate create and modify course proposals
* Can generate modify program proposals
* Can search for course and program proposals
* Can search, browse, and view approved courses and programs, as well as course/program version history
* Can copy existing courses and course proposals to a new course proposal
* Can manage course sets and learning objective categories
* Can generate dependency analyses

### Faculty/staff user

* Can perform generic user functions indicated above
* Can view course and program proposals that have been cancelled, withdrawn, rejected, or approved

### Admin user

* Can perform generic and faculty/staff user functions indicated above
* Can use administrator screens to create, modify, and retire courses
* Can use administrator screens to create and modify programs
* Can modify existing versions of courses and programs

### Workflow User: Can also be in one of the three previous categories

* Can review or approve course and program proposals for his/her organization, as detailed on Workflow Configuration and Associated Users
* Can view course and program proposals that are submitted to workflow, if the user is part of the proposal's workflow

### Sample Create Course Workflow, Roles, Permissions

| **Dept. Name** | **Subject Area(s)** | **Department Review** | **Division Review** | **School Review** | **Senate Review** | **Publication Review** |
| --- | --- | --- | --- | --- | --- | --- |
| Biology | BEES  BSCI  CBMG  MEES  MOCB  NACS | **Biology Dept. COC**  Chair: User One (user1)  Member: User Two (user2) | N/A | **College of Life & Chemical Sciences COC**  Chair: User Three (user3)  Member: user4 | **Senate COC**  Chair: Tester One (test1)  Member: Tester Two (test2) | **Publication Office**  Chair: Developer One (dev1)  Member: Developer Two (dev2) |
| History | His 1  His 2 | **History & Geog Dept.**   Chair: Test User 7 (testuser7)  Member: Test User 8 (testuser8) | **Behavioral Science Division**   Chair: Test User 9 (testuser9)  Member: Test User 10 (testuser10) | **College of Behavioral & Social Science**  Chair: Test User 5 (testuser5)  Member: Test User 6 (testuser6) | **Senate COC**  Chair: Tester One (test1)  Member: Tester Two (test2) | **Publication Office**  Chair: Developer One (dev1)  Member: Developer Two (dev2) |
| Geography | GEOG | **Geography Dept**  Chair: Test User 1 (testuser1)  Member: Test User 2 (testuser2) | **Social Science Division** Chair: Test User 3 (testuser3)  Member: Test User 4 (testuser4) | **College of Behavioral & Social Science**  Chair: Test User 5 (testuser5)  Member: Test User 6 (testuser6) | **Senate COC**  Chair: Tester One (test1)  Member: Tester Two (test2) | **Publication Office**  Chair: Developer One (dev1)  Member: Developer Two (dev2) |
| Music | MUED  MUET  MUSC  MUSP | **Music Dept**  Chair: User Five (user5)  Member: User Six (user6) | N/A | **College of Arts & Humanities** Chair: User Seven (user7)  Member: User Eight (user8) | **Senate COC**  Chair: Tester One (test1)  Member: Tester Two (test2) | **Publication Office**  Chair: Developer One (dev1)  Member: Developer Two (dev2) |

## PROPOSED SOLUTION MODEL

Verbat will be following a stand–alone fixed bid solution delivery model wherein the required solution for Phase One would be devised. Verbat’s solution architects have conducted a thorough research on the requirements and have come to the conclusion that our proposed solution, which is detailed further in this document, will meet the requirements put forth by the client.

**Key Strengths of Our Solution**

* Strong and Scalable platform accommodating to future enhancements.
* A framework which acts as a solution accelerator with building blocks that can be re-used in the future for building new components and features.
* Our light weight framework consumes fewer system resources thereby making the application perform faster.

## FUNCTIONAL REQUIREMENTS

The summary of features available in the proposed Curriculum Management Application are

1. **Course Management**
   1. New courses are created and existing ones modified or retired through a proposal process that allows users to collaborate on a proposal, and then submit the course for review. The system determines reviewers and the review path based on roles within the institution.
   2. Curriculum managers can use special administrative screens to create, modify, or retire courses. These streamlined screens allow users to manage the course inventory without using the standard proposal's curriculum review process.
   3. Courses may also be searched for and browsed.
2. **Program Management**
   1. Academic programs are created and modified by curriculum managers using special administrative screens. Undergraduate and graduate academic programs are defined by identifying descriptive information, requirements, learning objectives and governance and oversight.
   2. Existing programs also can be modified through a proposal process similar to course management, which supports collaboration, workflow routing, and approval.
   3. Programs may also be searched for and browsed.
3. **Course Set Management**
   1. Sets of courses can be defined and named for use in managing rules for courses (e.g. prerequisites) and programs (e.g. completion requirements).
4. **Learning Objectives Management**
   1. Learning objectives provide a way to manage the inventory of learning objectives. A learning objective is a statement that describes the knowledge, skills or abilities that a student would expect to gain by participating in a course or program. Curriculum Management provides tools to categorize, search, and re-use learning objectives.
5. **Dependency Analysis**
   1. Dependency analysis reports demonstrate relationships among curriculum components by showing the courses, programs, and course sets that use a selected course. This is particularly useful for understanding the potential impact of curricular changes.
6. **Organization and Workflow**
   1. Information about the people and organization within an institution is maintained through Identity Management (IM). The roles assigned to people through their affiliation and position in the organization defines the approval path for courses and programs.

The following sections of this document shall dwell into the detailed requirements entailed in the summary listed

## COURSE MANAGEMENT

Curriculum Management provides functionality for proposing, approving, modifying, and activating courses that can then be published in a course catalog. Curriculum managers can also retire courses. What information is specified for a course is fully configurable by the institution.

### Course Information

Curriculum Management provides the means to record the details of a course. You can configure Curriculum Management to present and require data fields based on your institution’s specific business requirements. The following is a summary of the information that is included in KS Curriculum Management as it is delivered:

1. **Course Information**
   1. General information about a course, including the title (with options for displaying differently in the catalog or transcripts), instructor(s) for the course, the course description and the rationale for proposing a new course or modifying an existing one
2. **Governance**
   1. The organizations responsible for oversight and administration of the course and the campus locations that will offer the course. The selected value for the Curriculum Oversight field will determine what organizations need to be involved in the review and approval of the course. Based on this information, Curriculum Management determines the approval path for the course creation or modification proposal.
3. **Course Logistics**
   1. Specifics such as in what term(s) the course is typically offered, the duration of the course, how grades and assessments will be conducted and recorded for the course, whether a student can audit the course, the course format (lecture, lab, etc.), and the outcome of the course such as credits awarded.
4. **Learning Objectives**
   1. Identifies the expected outcomes the student will achieve upon successful completion of the course. Categories can be assigned to learning objectives in order to reflect institutional taxonomies or classification schemes.
5. **Course Requisites**
   1. Course requisite rules identify requirements for enrolling in the course, such as specific courses, previous credits, enrollment in a program, grade-point averages, etc. Co-requisites, Recommended Preparation, Anti-requisites, Courses that Restrict Credit, and Courses Repeatable for Credit also can be defined.
6. **Active Dates**
   1. Specifies the term during which the course will become active for students to enroll, whether the course is a one-time/pilot course, and, if applicable, the term for which the course will no longer be offered.
7. **Financials**
   1. Identifies any fees associated with the course as well as the organizations associated with revenue generated and expenditures incurred by offering the course.
8. **Authors and Collaborators**
   1. Identifies specific people not on the standard approval path who can edit, comment or view the course proposal.
9. **Supporting Documents**
   1. Allows additional files, such as a syllabus, to be attached to the course.

#### Course Requisite Rules

Course requisites are structured rules that define a student’s eligibility to enroll in a course. What requisites are available for selecting can be configured to suit an institution’s specific needs. Examples of requisites that shall be defined in Curriculum Management as delivered include:

* Courses the student must have completed before enrolling in this course.
* Minimum GPA or specific grades in another course or set of courses.
* Enrollment in a specific program.
* Previously completed courses that would make a student ineligible to take this course.

The rules are defined using drop-down lists to construct structured rules statements. These rules statements can then be used by an online enrollment application to determine if a student is eligible for a course.

### Course Processes

There are a number of activities that a course navigates throughout its lifetime. Curriculum Management provides the following processes by which to manage courses:

1. **Create a Course**
   1. Someone at the institution proposes a new course, supplying the necessary information, such as requisites, instructors, times offered, financial information, etc. The person who initiates the proposal can identify collaborators who can also edit or review the course information. Once submitted, KS Curriculum Management routes the proposal to the appropriate people based on their roles within the institution.
   2. Proposals can be created from scratch or can be copied from existing approved or proposed courses. Copied proposals include pre-populated data from the source course. The fields that get copied are configurable by an implementing institution.
   3. Curriculum managers can be authorized to access special administrative screens to create a course. Administrative screens are streamlined for faster data entry, allow the user to hide non-required fields, and do not require proposed courses to go through the standard review and approval process.
2. **Modify a Course**
   1. An approved course may be modified through the proposal process similar to a new course.
   2. Curriculum managers can also be authorized to access special administrative screens to modify a course. Using these special screens, the curriculum manager can choose whether to modify an existing version of the course or to create a new version, and the modification is not required to go through the standard review and approval process.
3. **Retire a Course**
   1. An approved course may be retired through the proposal process, indicating important information like the last term in which the course can be offered and retirement rationale.
   2. Curriculum managers can be authorized to access a special administrative screen to retire a course. Using this streamlined screen, the curriculum manager can retire a course and is not required to go through the standard review and approval process.

## PROGRAM MANAGEMENT

Curriculum Management provides functionality for creating, modifying, approving, and activating academic programs (under graduate, graduate, post graduate and doctoral). What information is specified for a program is fully configurable by the institution.

### Program Information

Curriculum Management provides the means to manage the details of a degree program. You can configure Curriculum Management to present and require data fields based on the institution’s specific business requirements. The following is a summary of the information that is included in Curriculum Management as it is delivered:

#### Key Program Information

Information about a program such as the title, degree type, the title of the program (with alternatives for transcripts and diploma), the term during which it will first be offered and campus locations where it is offered.

#### Program Managing Bodies

What organizations are responsible for oversight and administration of the program. To modify program proposals, the Curriculum Oversight board shall determine what organizations need to be involved in the review and approval of the program. Based on this information, Curriculum Management determines the approval path for the program modification proposal.

#### Specializations

Sub-programs within a given program that can be either optional or required to complete the program. All of the information collected for a parent program can be collected for a child specialization. The system verifies that certain specialization information, such as managing bodies and start/end terms, is valid when compared to the parent program.

#### Description and Catalog Information

A description of the program, including alternative text for publication in the catalog, the duration of the program (four year, two year, etc.) and the core faculty members for the program.

#### Program Requirements

Requirements for entering the program, progress requirements for satisfactory performance to remain in the program, and requirements for completing the program.

#### Learning Objectives

Identifies the expected outcomes the student will experience upon successful completion of the program. Learning Objectives can be specific to the program or they can be standard learning objectives that are assigned to more than one program.

#### Supporting Documents

Allows additional files, such as a syllabus or letter, to be attached to the course.

### Program Processes

There are a number of activities that a program navigates throughout its lifetime. Curriculum Management provides the following processes by which to manage programs:

#### Create a Program

* A curriculum manager can use special administrative screens to create a new program, supplying the necessary information. The program can then be approved and activated by the curriculum manager.
* Note that the program creation process does not utilize a proposal process that utilizes review and approval workflow like the course creation process.

#### Modify a Program

* Someone at the institution proposes changes to a program, supplying the necessary information. The person who initiates the proposal can identify collaborators who can also edit or review the course information. Once submitted, KS Curriculum Management routes the proposal to the appropriate people based on their roles within the institution.
* Curriculum managers can be authorized to access special administrative screens to modify a program. The curriculum manager can choose whether to modify an existing version of the program or to create a new version.

## COURSE SET MANAGEMENT

Curriculum Management allows you to create groups of courses that can then be referenced by other processes. Course sets are used when creating course requisite rules or program requirements. The following are example uses for this feature:

* Course Requisite Rules: Multiple Choice Prerequisites

A set of courses from which a student must take one, but not all, to be eligible to enroll in a subsequent course.

* Program Requirements: Core Courses

A defined set of courses that all students, or all students within a program, must complete.

Course sets may be fixed or dynamic lists of courses. A fixed course set is defined by explicitly adding courses to the list by course number (MATH101, MATH102, MATH103, etc.). A dynamic course set is defined by applying some criteria, such as any 100-level course in the Mathematics department. As courses are added that meet the criteria, they are automatically added to the dynamic course set.

## LEARNING OBJECTIVES MANAGEMENT

Learning objectives provide a way to manage the inventory of learning objectives. A learning objective is a statement that describes the knowledge, skills or abilities that a student would expect to gain by participating in a course or program. Curriculum Management provides tools to categorize, search, and re-use learning objectives.

The key principles of a learning objective include:

1. It is specific.
2. It is measurable/observable
3. It is attainable
4. It is relevant and results-oriented
5. It is targeted to the learner and the desired level of learning.

Learning objectives can range from a simple statement describing the student’s expected mastery of some aspect of the subject to a specific certification. A learning objectives can be attached to any number of programs or courses that have that objective as an expected outcome.

**Learning Objective Hierarchies**

* Learning objectives can be maintained in hierarchies with levels and sublevels if desired. Learning objectives have various levels of abstraction and specificity and systems have evolved for describing the hierarchies of learning objectives (e.g., Fink’s Taxonomy and Bloom’s Taxonomy). An institution can decide wither they are interested in using taxonomies to manage the relationships between learning objectives.

**Learning Objectives Categories**

* Since many learning objectives are short, free-form statements, categories provide a means for authors to provide more information about their learning objective such as accreditation categories, skill-based categories, and subject-matter categories. These categories facilitate discovery of learning objectives but also enhanced reporting capabilities for the institution.

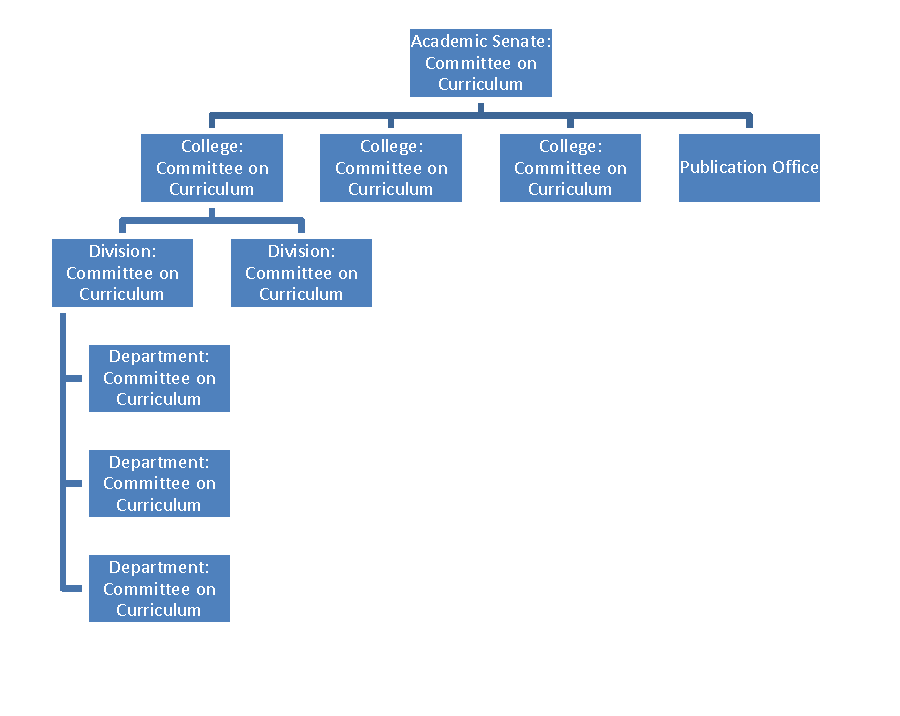
## ORAGANIZATION, PEOPLE and ROLES MANANAGEMENT

The organizational structure of the institution and the specific people and their roles within that organization, are the keys to defining the workflow processes through which course and program changes are routed. Curriculum Management uses Identity Management (IM) and Enterprise Workflow (EW) to determine who can propose, review, modify and approve courses and programs. These responsibilities are not tied directly to individuals, but rather to roles that individuals occupy in the organization.

For example, if a new course is proposed for the Biological Sciences department, Curriculum Management determines the approval path for that course based on what it knows about the organizational structure. The first step may require review and approval by the Chair of the Biological Sciences department’s Committee on Curriculum Once approved, the proposal moves on to the institution’s Committee on Curriculum. Curriculum Management determines the current members and chair of the committee and presents the proposal to them for review, comment or approval. After the appropriate approvals have been obtained, Curriculum Management determines the next step is to present the proposal to the Publication Office for final approval.

As peoples’ roles change, Curriculum Management will be able to react to these changes and route the proposal to the appropriate people. For example, if a member of the Committee on Curriculum is replaced with a new member, Curriculum Management will present proposals to the new member instead. By keeping the organizational structure, people and roles current, an institution can ensure that course proposals are routed to the appropriate people.

You can configure IM and EW to match your institution’s specific needs. Without modification, Curriculum Management assumes the following organizational structure and roles for managing workflow.



## WORKFLOW STATES

| **Workflow Status** | **Description** | **Proposal States** |
| --- | --- | --- |
| **Saved** | Document has been created by author prior to routing | Saved |
| **Processed** | Document has received all required Approvals | Approved |
| **Final** | Document has received all required Approvals and has all Acknowledgments cleared; proceed to post-processing | Approved |
| **Exception** | Issue/problem with routing; document requires intervention by Exception User/Workgroup | Exception |
| **Enroute** | Document is being routed | Enroute |
| **Disapproved** | Document has been Disapproved by reviewer; routing stops | Rejected (by Reviewer), Cancelled (by Initiator), Withdrawn (by Initiator) |

## PROPOSAL STATES

**PROPOSAL STATES:** Proposal states *are* under the purview of KS and, to the extent possible, are aligned with workflow status.

| **Proposal States** | **Description** | **State set when:** | **Workflow Status(es)** |
| --- | --- | --- | --- |
| **Saved** | The **proposal** has been saved but not routed | The proposal is first Saved by the initiator | Saved |
| **Cancelled** | The **proposal** has been saved, but the initiator wants to cancel the proposal instead of submitting | The proposal is Cancelled by the initiator | Disapproved (by Initiator) |
| **En-route** | The **proposal** is being routed | The proposal is Submitted to workflow | En-route |
| **Withdrawn** | The **proposal** has been withdrawn by initiator prior to final approval | The en-route proposal is Withdrawn by the initiator | Disapproved (by Initiator) |
| **Approved** | The **proposal** has received all required Approvals | The last reviewer the on Route Log with an Approve Action Request approves the proposal | Processed, Final |
| **Rejected** | The **proposal** has been rejected | A reviewer on the Route Log with an Approve Action Request disapproves the proposal | Disapproved |
| **Exception** | There is an issue with the routing | Routing fails | Exception |

## COURSE STATES

| **Course States** | **Description** | **State change expected when:** | **Allowed Actions** | **Valid Next States** |
| --- | --- | --- | --- | --- |
| **Active** | **course** has been prepared for offering (i.e. reviewed and updated by a curriculum administrator to ensure course data is consistent with institutional policies and practices) | course proposal has been approved at Publication Office node of reference workflow | Modify with Version  Modify without Version (for admin screens only)  Suspend  Retire  Copy to New Proposal | Draft  Suspended  Superseded  Retired |
| **Approved** | **course** has been approved (i.e. completed curriculum review business process) | Reference implementation will not utilize Approved state. Courses will move directly from Draft to Active. An implementing institution could have a two-step process in which courses are first Approved, then Active. | Reference implementation will not utilize Approved state | Reference implementation will not utilize Approved state |
| **Draft** | **course** is in the process of being proposed/created | course proposal is saved OR   course is entered into system via screen(s) | Initiator:  Cancel  Submit  Withdraw  Copy to New Proposal   Reviewer:  Approve  Reject  Return to Previous Node  Blanket Approve   Admin:  Modify without Version (to be implemented for admin screens only) | Not Approved  Active |
| **Latest States** | Grouping of the latest states. Cannot have more than one version among these states: active, suspended, & retired. Default search result should include just these versions as well. |  |  |  |
| **Not Approved** | **course** creation or modification was not approved | Course proposal cancelled or withdrawn by initiator  OR rejected by reviewer. | Copy to New Proposal | None |
| **Retired** | **course** has been approved for retirement | retire course administrative action is approved   Retire Course | Modify with Version  Modify without Version (for admin screens only)  Copy to New Proposal | Draft |
| **Superseded** | **course** has been superseded by newer version | set via batch job when newer version becomes active | Modify without Version (for admin screens only) | None |
| **Suspended** | **course** has been approved but is temporarily not eligible for offering | Suspended | Activate  Modify with Version  Modify without Version (for admin screens only)  Retire  Copy to New Proposal | Active  Draft  Retired |

## COURSES and PROPOSAL WORKFLOW with STATES



# APPLICATION SERVICES

## Academic Time Period Service

### Description

The Academic Time Period Service manages Academic Time Periods (ATPs) and their associated Milestones. The service provides a flexible but structured way to define the various time frames that are used throughout the definition, offering and scheduling of Learning Units. This is a catalog service with basic operations.

Milestones are a single date/time or a range between two date/times. A Milestone can represent an event such as a deadline for application submission (single date) or a period such as Grade Submission (date range). Milestones are managed independently from an ATP and may be mapped to one or more ATPs. There is not an explicit service entity for the ATP - Milestone relation.

An Academic Time Period (ATP) has a start date/time and an end date/time. These values do not constrain the related Milestones. In other words, an ATP for Fall semester may have a pre-registration Milestone with a date in May.

### Key Concepts

* Milestones are independently managed and mapped to ATPs. The relationship is implied directly through service operations and explicit relationship entities.
* A Milestone with a day but no time component has an attribute of "all day."
* A Milestone with a "range" attribute requires an end date, null not allowed.
* ATP relationships capture the concept of nested Terms and mapping of Terms to Academic Calendars.

## Canonical Learning Unit (CLU) Service

### Description

The Canonical Learning Unit Service supports the management of Canonical Learning Units (CLUs). This includes the development and approval of new Learning Units and substantive changes to existing Learning Units. A Learning Unit (LU) is any learning-related activity that needs to be tracked by the institution or the learner. Examples of learning units include the traditional curriculum of courses and degrees, professional and extension programs, and non-academic activities such as leadership development and service learning. All credit and non-credit learning activities, whether traditional or non-traditional, are learning units.

Learning Units (LU) are first created at the canonical level as Canonical Learning Units (CLUs). When a CLU is offered during a Term (ATP), it is referred to as a Learning Unit Instance (LUI). LUIs are created and managed in the Learning Unit Instance Service. For example, an LU that represents a Course is created, proposed, and approved as a CLU and offered for a specific ATP as a LUI.

The CLU service is broad and covers many aspects of learning either directly or by reference (other services). The focus of this service initially is the creation of the inventory of CLUs that comprise an institution's offering including the associated learning objectives (outcome), learning results, supporting documents, evaluations, statements (structured rules logic specific to CLU), resource types and academic time periods.

### Key Concepts

**Canonical Learning Unit Types (CluType)**

Provide the basic structure or template for creating the CLU. Depending on the type, there may be different information associated with the CLU. Examples of CluTypes include Course, Program, Project, and Assessment. The granularity or CluType depends on the implementation.

**Canonical Learning Units (CLU)**

* This is the inventory of Learning Units (LU) for an institution, the collection of LU that have been defined.
* These have generally been reviewed and approved through an approval process.
* There is a status (e.g., active, inactive, retired) associated with the "canonical" learning unit.
* CLUs are often approved for a range of Academic Time Periods.
* There are several ways to compose CLU into complex structures. LUI inherit the composition of the CLU.
  + CLU-CLU Relations can be created to connect CLUs. The relation has a LU-LU RelationType which is constrained by the CLUType. This construct supports composite CLU, for example, a Chemistry class that is comprised of a Lecture and a Lab. CLU-CLU Relations are used to define a series of CLU where all courses must be completed before credit is awarded.
  + CLU Sets are collections of CLU. They can be dynamic (based on a query) or a simple enumerated lists. CLU sets are unordered sets of CLU.

**Learning Unit Instance (LUI)**

The specific offering (occurrences) of a CLU typically during a specific Academic Time Period (ATP), though in some instances there may not be an associated calendar dates (e.g. online courses). These are managed in the Learning Unite Instance Service.

**Learning Objectives (LO)**

* These objectives are the intended outcome(s) for the LU.
* LO are separate and related many-to-many to CLU, so a single CLU has multiple learning objectives and the same learning objective may be related to several CLUs.
  + There is a separate Learning Objective (LO) service that acts as a catalog service to manage the various LO that may be related to CLU.
  + LO may also be related directly to a Student, not part of this service (Current thinking is that this would be part of the Learning Plan (LP) which may be subsumed by the TBD Student Service).

**Result Options**

* The wrapper for the result components defined within the Learning Result Catalog service.
* Each result option is associated with a specific result option usage type.
* CLUs and LUIs will not attach directly to result options.

**Result Sets**

* The collection of one or more result options.
* The options in a set represent one permutation of results that can be offered by a CLU/LUI and achieved by a student.
* Each Result Set is associated with an LU Type to constrain which CLUs they can be attached to.

**CLU Results**

* The association of CLUs to defined result set.
* The CLU Results represent the possible sets of learning results available for LUI instantiation.
* The CLU Results instantiated for a LUI in turn represent the available results for LPR.
* LU Statements can be evaluated during creation of an LPR in order to determine which results sets are valid for a specific person.

**Documents**

* The associated supporting materials that accompany the CLU from inception, through approval, catalog listing, offering, evaluation and finally retiring.
* The CLU service manages the connection to documents but not the actual maintenance and storage of the documents themselves.
* They have been separated from CLU-Info in support of different authorization access.

## Comment Service

### Description

The Comment Service allows for the creation and management of user comments and tags associated with other objects across the system. There is no expectation that the objects know anything about the tags or comments; therefore objects can be deleted from the system even though tags or comments referencing those object may exist.

### Key Concepts

**Comment**

* A comment is a block of text that can be associated with certain objects often as part of a collaborative or workflow process.
* The comment type can be used to help the application differentiate comments that might need to be displayed in particular contexts, such as during Course Approval.

**Tag**

* A kind of metadata which helps categorize or describe an item and may allow it to be found again by browsing or searching.
* Each tag is contains of a multipart key consisting of a namespace, a predicate and a value.
* User profile-specific tags can be accomplished with the existing tag construct, e.g., namespace for a principal, etc.
* The tag type will likely only be a single value for tag type.
  + The justification for adding it is to maintain consistency in being able to describe the structure with the dictionary operations.

**Reference**

* A pointer to an object that exists elsewhere in the system, e.g., a CLU, a document, etc.
* The reference type indicates the type of object that is referenced by the comment or tag and is part of a composite id/type key for the reference.

## Course Service

### Description

A course is a specified area of knowledge contained and taught independently from other areas of knowledge that may or may not be related. Courses typically have specific learning objectives, defined student learning outcomes and assessments. They can be recognized for academic credit or not. Courses can be delivered in various instructional formats such as lecture, lab, discussion, seminar, colloquium, etc. A course would utilize at least one instructional format, but often will use more than one.

Course is pre-configured type of CLU (Canonical Learning Unit) that represents a credit course. The course Info object supports a number of complex course options – cross-listed, joint and variations. Course can have any number of associated learning objectives and requisites (rules that model pre-, co-, anti-requisites and enrollment restrictions).

### Key Concepts

**Course, Format and Activity**

* Each course has one or more formats, each of which can have one or more activities.
* The activity represents the specific teaching experience through which learning is imparted to the student (i.e. Lecture, Lab, etc).
* The format describes the "footprint" of activities that describe the course.

**Financials**

* Financial information is associated with the course and not the format or activity. It includes the fee justification, list of fees, associated revenues and expenditure (org) distributions.

**Grading and Credits**

* Grading and credits are associated with the course and not the format or activity.
* Multiple grading (e.g., letter, pass/not pass) and credit (fixed/variable with units) options can be specified.

**Complex Courses**

* Any number of joint courses (code, ID and description) can be specified for a course; these courses must already exist.
* Cross-listed courses are modeled as alternate IDs related to the course entity; multiple cross-listed IDs can be specified.
* Variations are modeled as additional "suffixes" including the variation code and the related title.

**Learning Objectives**

* Multiple Learning Objectives can be associated with courses. These can be structured, from a managed catalog, or added (and reused) ad hoc as part of defining each course.
* The actual management of the LO is managed in a separate Learning Objective service. The course service provides the mechanism for associating LOs to the course.

**Course Requisites**

* Multiple Course requisites, e.g., Pre-requisites and enrollment restrictions, can be associated with courses.
* The actual management of the requisites is managed in a separate Statement service. The course service provides the mechanism for associating requisites to the course.

## Document Service

### Description

The Document Service supports the management of document objects. Relations between stored documents and external entities are managed through the respective entity service.

### Key Concepts

**Document Type**

* The document type indicates the format of the document, e.g., text, jpeg, xml
* The usage of types within this service is similar to MIME types

**Document Category**

* Categories are used in this service for classification and organization of documents
* Upon creation, the document must be associated with a category
* Additional categories can be added or removed as long as each document is associated with at least one category

**Retrieving Document Information**

* The "Read" operations with the service are used to retrieve document(s) when the document identifier is known
* Output data includes both the meta-information and the document binary
* In cases where the service caller is interested in retrieving custom field sets using criteria other than the document identifier, custom search results and search types can be configured for the operation

## Enumeration Management Service

### Description

Enumeration Management service is only accessed by authorized callers configuring some piece of the system. The *code* identifies the value within an enumeration.

### Key Concepts

* This service only supports retrieval of enumerations which are managed through this service.
* Only manually maintained enumerations are supported by this service; enumerations which act as views of configuration information or views of existing objects are maintained by the service of record.

## Learning Objective Service

### Description

The Learning Objective Service supports the management of formal learning objectives. This is primarily a catalog service with basic operations.

### Key Concepts

**Learning Objectives in a Repository**

* A Repository exists to provide a way to enable different policy and authorization rules to be established. For example, you might have one repository for externally maintained LOs, and another for your institutions. The external ones probably are only maintained by a couple of individuals as the external source make changes, but you can have an entirely different set of rules around your local LOs.
* It is not mandatory that the LOs are all arranged in a tree, it is possible to have a flat repository which has no root.
* A learning objective is attached to only one repository, and can optionally be at the root or as a relation with another learning objective. For an existing learning objective, it can be moved to another node or attached to multiple LOs within the same repository.

**LO LO Relations**

* A relation between LOs and other LOs e.g., supports/in support of, is equivalent to, etc.
* This is a directional reference that can be used between learning objectives in the same repository or different repositories.
* For a bi-directional equivalency, the equivalency reference must be individually established on the learning objective relation type (using both name and revName fields on the loLoRelationTypeInfo structure).

**Learning Objective Categories**

* Categories can be created and attached to a single learning objective repository as a means for grouping learning objectives within the same repository.
* Learning objectives can be associated with one or more categories within the same repository.

## Learning Result Catalog Service

### Description

The Learning Result Catalog service supports the management of learning results. It provides the basis for defining the result types and values that can be associated with the catalog LU (CLU), the offered LU (LUI), and the specific student (LPR).

### Key Concepts

**Result Values Group**

* Result Value Group provide a domain of result values, while acting as an abstract result concept. These domains may correspond to scales or simply sets of mutually exclusive achievable results.
* The Result Value Group typically provides context if it maps to a scale, but in other cases, such as credit, the Result Value Group provides little to no additional information, other than acting as a known grouping of Result Values.
* Values associated with Result Values Groups are limited to the same type.

**Result Values**

* Result Values serve as a container to the value of a specific result type.

**Specific Result Type**

* Specific Results are the foundation of all learning results known by the service.
* The types of results currently defined are credentials, credits, and grades.
* New types of results can be added to this service as needed.
  + *Credentials* correspond to degrees or certificates.
  + *Credits* correspond to an assessment of the amount of time or content. For example, Carnegie Units might be captured using this concept, with the number of units retained within the value field of the creditInfo structure.
  + *Grades* correspond to quantitative assessment of one's progress and mastery of the material covered in the course. Grade values belong to a single scale to allow for easy scale to scale comparisons. This concept is distinct from what might be represented in the Result Value Groups, as the domain of potential grade values may include grade values which are orthogonal from the standard assessment. For example, most current systems have "grades" corresponding to incomplete or withdrawn, which are not true assessments and thus comparisons.

## Message Service

### Description

Manages locale specific messages for internationalization. A *message* is unique in the context of *locale* and *group*. The *locale* parameter allows for operations pertaining to messages whose locale is different from that of *ContextInfo* locale information.

### Key Concepts

* MessageGroupKey is used to group messages into categories such as business concepts, organization, program, course, or overall categories such as common and validation.
* Locale is part of contextInfo which is specified as part of virtually every service call.

## Organization Service

### Description

This service manages organizational units that have a relationship to the institution. The organizations may be internal and include officially recognized organizations (e.g. Departments, Faculties, Schools) or unofficial organizations (e.g. clubs or student groups), or they may be external organizations (e.g. companies, other institutions, government, associations). This service also manages the relationships between people and organizations.

### Key Concepts

* Organizations are distinguished from authorization groups in that organizations deal directly with people while groups deal directly with principals. In other words, organizations may be comprised of individuals who have no way to authenticate themselves (and thus have no unique permissions) and AZ groups may have principals which are linked to non-human entities (such as batch jobs, other services, etc.).
* Organizations and groups may be related, in that a member of an organization may have one of their principals associated with an AZ group, but this is not required.

## Program Service

### Description

A program is a prescribed grouping of learning units such as courses, activities, competencies, learning objectives, and/or institutional/core requirements that lead to a recognized body of knowledge. Programs may also include requirements as to the order, timing, performance, and results (e.g. GPA) of the grouping of learning units that make up a program. The end result of a completed program may be an acknowledged level of accomplishment (e.g. a major, minor or concentration) or a credential (e.g. certificate, degree, etc.)

### Key Concepts

**Credential Program**

A classification of canonical learning units (types) that lead to some sort of credential. Here are some of the types of credential programs.

**Major Discipline**

* An organized curriculum offered as a major area of study recognized as an academic discipline by the higher education community as part of a baccalaureate degree program.
* A learner may complete more than one major area of study within a single baccalaureate program.

**Minor Discipline**

* Any organized curriculum that is offered as part of an individual student's degree program and which enhances or complements the degree to be awarded in a manner that leads to specific educational or occupational goals.
* The credit hour length is set in accordance with institutional policy and typically requires less coursework/study than a major area of study.

**Program Variations**

* Variations of a major area of study that lead to specific educational or occupational goals, also known as Specializations, Concentrations, Areas of Emphasis, Tracks, etc.
* Some variations may represent a portion of unique curricula in relation to the major area of study, while others specify more focused paths of study within the boundaries of a major area of study.

**General Education or Core Program**

* An organized grouping of courses that is required for all students as a foundation of general knowledge.
* The courses typically span a breadth of knowledge in areas such as Mathematics, English Composition, History/Social Science, Humanities and Fine Arts, and Natural Sciences.

**Honors Program**

* Departments often have Honors programs that are managed at the department level and are available to the major programs that department offers.

**Program Requirement**

* Programs are typically defined by the "requirements" which describe program goals and the courses that can meet those goals.
* These are defined in CM as "rules" which are grouped by statement type to define requirements for Entrance, Satisfactory Progress and Completion.

## Proposal Service

### Description

The Proposal Service supports the management of Proposals. The proposal is general and can be associated with any arbitrary entity in the system, for example, a learning unit. Proposals have types that capture the activity being proposed, for example, creating a new course or Updating the prefix for a set of courses or Launching a new student club.

A proposal may reference more than one (1) object, provided that the objects are all of the same reference type and that the other data contained in the proposal applies to all of the objects referenced. An object (e.g. CLU) is involved in **only one** active proposal at a time.

### Key Concepts

**Proposal Types**

* Proposal types provide the basic structure or template for creating the Proposal, e.g., Create New Course, Course Correction.
* Depending on the type, there may be different information associated with the Proposal. Examples of Proposal Types include Course Correction, Course Modification, etc.
* The granularity or Proposal Types depends on the implementation.
* The type of the Proposal may also indicate the type of CRUD activity and have implications for the associated object. For a proposal type of Course Correction, the referenced object is an existing CLUID; for a proposal type of Create New Course, it would be a new CLUID.
* Proposal Type is constrained by Reference Type.

**Reference Type**

* Reference type is overall type of the Proposal e.g., Create New Course, Course Correction.
* Reference type along with an ID, reflects the specific entity associated with the proposal. For example, a Proposal may reference an object of type CLU along with its CLU Id. The Reference Type Key and the Reference ID concepts are also found in the Comment Service.

**Proposal Doc Relation Type**

* Proposal document relation types reflect the relation type between a proposal and a document, e.g., proposal rationale, syllabus, etc.
* While it could be similar to the luDocRelationType, ultimately these are two distinct concepts. It is also distinct from what the documentTypeKey which captures the file format, e.g., PDF.

## Search Service

### Description

The Search Service supports the creation of typed searches that query the underlying data store and return results in tabular form. Conceptually it can be viewed as a wrapper for predefined (typed) SQL statements. It has two main purposes:

1. The definitions of searches to meet specific purposes without modifying the contract.
2. The definition of searches that return data other than objects and their attributes, for example counts or sums.

### Key Concepts

**Search Type**

* Equivalent to a named search or parameterized query. This describes the inputs as well as the outputs from calling a search of this type. Since the criteria and results are uniquely named, this can be seen also as the combination of those two concepts.
* Search types provide a handle to both the criteria to be provided and the results to be returned.
* A user may be authorized to be able to perform a type of search even if they are not authorized in the general case to view the more complete information of the entities involved. This provides something akin to view functionality, albeit in a constrained fashion as the results have been flattened.
* The search results returned may also be dependent on underlying authorization. In other words, certain searches may limit the returned values by the granted authorizations.

## State Service

### Description

State service is used for managing the following: valid States of an object, constraints associated with a state change and propagation triggered by a State change. It also describes a way to manage constraint and propagation of States.

### Key Concepts

**State Event Features**

* *State change* from the current object state to another state triggered by an *event*/*reason*, subject to desired constraints.
* *State Propagation* – an object's state change could trigger state changes in other objects, subject to desired constraints.
* *Cross Constraints* – an object's current state may constrain the valid states of another object, subject to desired constraints.

**Lifecycle**

* State service organizes collections of States into a Lifecycle.
* A Lifecycle indicates the type of entity to which a set of States apply. For example, a LUI may go through a lifecycle for instantiation with States of "rolled over," "scheduled," and "published."
* The "instantiation lifecycle" is only applicable to LUIs although the service contracts accept any identifier.

**State Propagation and Constraints**

* Events, implemented as the type of State Change object, can be used to capture the reason for a state change. For example, 'Professor left' and 'Low enrollment' could be two different events leading to the same Course Canceled state.
* Allow transition from any state to any state, and use 'constraints' as needed. Constraint operators are:
  + ALL: all related objects must be in a State
  + EXISTS: at least one of the related objects must be in a State
  + NONE: none of the related objects must be in a State

## Version Management Service

### Description

The Version Management Service inspects versions of an object. The Version Management Service is used in conjunction with another service to examine the version history of an object. This service provides access to the identifiers to retrieve older or future versions of objects from their respective services.

The Version Management Service is only useful when used in conjunction with another service that supports versioning.

### Key Concepts

**Terminology**

* Versioned object: an object at a specific version.
* Current object: an object that is at the current version. The current version may not be the latest version.
* Draft object: an object that is at a higher version level than the active object.
* Version Specific Id: an identifier that identifies a particular version of an object.
* Version Independent Id: an identifier that always refers to the object at the current version. It is independent of the version.

**Id Management**

Typically, every object is identified by a unique Id. A service that supports version management identifies each version of an object by its own unique Id. This is the version specific Id. A version specific identifier is assigned to an object for each new version created.

When a version specific identifier is used within a service that supports version management, the operations and relationships refer to a specific version of that object. When new versions of the object are created, the version specific identifier continues to refer to the older version.

A single version of an object is designated as the current version. The current version of an object is identified by a version independent identifier. The version independent identifier is assigned when the first version of an object is created and never changes as new versions of an object are created.

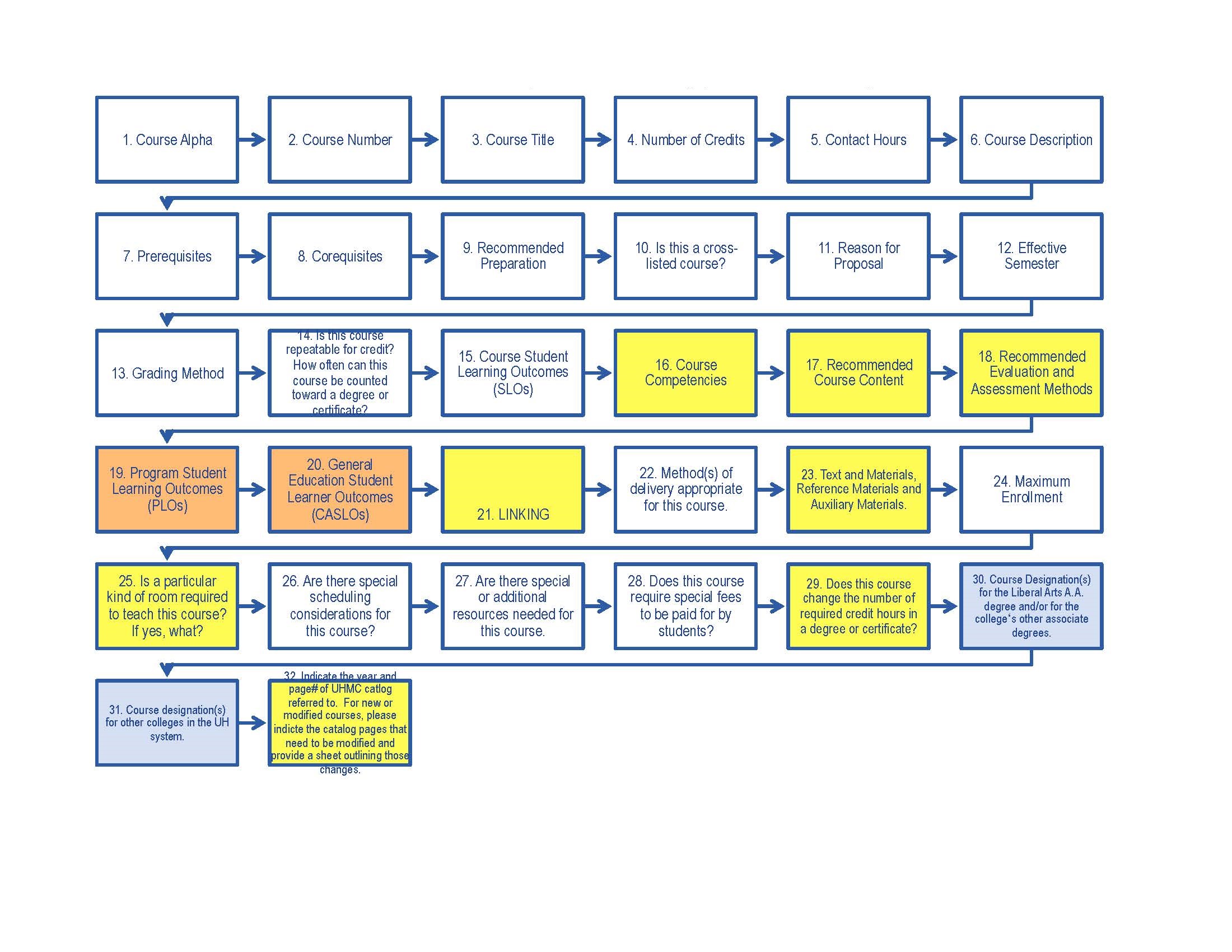
When a version independent identifier is used within a service that supports version management, the operations and relationships refer to the current version of that object. When the current version of an object moves forward to the next version, the operations and relationships refer to the next (now current) version.

# DETAILED REQUIRMENT SPECIFICATIONS

|  |
| --- |
| **Courses** |
| **Course Creation** |
| Create course through proposal process |
| **Create** a Course **through administrative screens** |
| **Course Definition** |
| Define course Key Identification information |
| Define course description |
| Define alternate identifiers, cross-listing and version codes |
| Specify join offerings |
| Specify key dates and terms |
| Identify course resources (Faculty) |
| Identify other course resources |
| Define multiple formats and activities |
| Define course grading and credit options |
| Define learning objectives |
| Define course requisites |
| Define organizational relationships |
| Specify financial information |
| Attach supporting information |
| Attach comments |
| **Course Management** |
| Modify course through proposal process |
| Modify course through administrative screens |
| Retire a course |
| **Course Versioning** |
| Version course as a result of modification |
| Prevent versioning because of modification |
| Activate an approved course version |
| Ability to publish approved course version in centralized catalog |
| **Course Search** |
| Browse catalog for course by subject area or organization |
| Browse catalog for course using other attributes |
| Ability to search for coarse |
| View a course |
| Select multiple views of course |
| View version history of course |
| Ability to print course |
| Ability to export course to multiple formats |
| **Programs** |
| **Create Program** |
| Create program through proposal process |
| Create a program through administrative screens |
| **Program Definition** |
| Specify key program identification |
| Specify program description |
| Specify program dates and terms |
| Identify program resources - faculty |
| Define program learning objectives |
| Ability to define Program Entrance, Satisfactory Progress and Completion Requirements |
| Ability to specify Organizational Relationships |
| **Manage Program** |
| Ability to modify a Program through a proposal process |
| Ability to modify a Program through administrative screens |
| Ability to retire a Program through a proposal process |
| Ability to retire a Program through administrative screens |
| **Program Versioning** |
| Ability to version a Program as a result of a modification |
| Ability *not* to version a Program as a result of a modification |
| Ability to activate an approved Program version |
| **Program Publishing** |
| Ability to publish approved Program versions in a centralized catalog |
| Ability to designate *other*publishing targets  (i.e., not the catalog) |
| Ability to define information to be published by target |
| **Browse and search programs** |
| Ability to browse the catalog for a Program |
| Ability to search for a Program |
| Ability to view a Program |
| Ability to view the version history of a Program |
| Ability to print details of a Program |
| Ability to export details of a Program |
| **Proposals** |
| **Start Proposals** |
| Ability to start a Proposal to create a Course from blank |
| Ability to start a Proposal to create a Course from a copy of an another approved Course |
| Ability to start a Proposal to create a Course from a copy of an another proposed Course |
| Ability to start a Proposal to create a Course from a template |
| Ability to start a Proposal to modify a Course from a copy of the Course |
| Ability to start a Proposal to modify a Program from a copy of the Program |
| Ability to save and return to a Proposal |
| Ability to cancel a Proposal |
| **Proposal Collaboration** |
| Ability to attach Supporting Documents to a Proposal |
| Ability to add Collaborators to View, Comment and/or Edit a Proposal |
| Ability to notify Collaborators that their participation is requested |
| Ability to remove Collaborators from a Proposal |
| **Proposal Review** |
| Ability to specify a workflow based on Proposal type |
| Ability to specify a workflow based on Proposal data |
| Ability to submit a Proposal to workflow |
| Ability to withdraw a Proposal from workflow |
| Ability to assign reviewers View, Comment and/or Edit proposal permissions |
| Ability to Request a Decision from a reviewer |
| Ability to notify reviewers of a pending review/decision request |
| Ability of reviewers to Enter a Decision on the proposal |
| Ability of reviewers to determine the status of a proposal |
| Ability to view the History of the proposal as it moves through the review process |
| Ability to record the final Outcome of the review process |
| Ability to review grouped proposals simultaneously |
| Ability to group proposals |
| **Archive, Search & View Proposals** |
| Ability to archive a proposal |
| Ability to Search for Proposals |
| Ability to view a Proposal |
| Ability to export details of a Program |
| Ability to print a Proposal |
| **Course Set** |
| **Create Course Set** |
| Ability to create a fixed or dynamic Course Set |
| Ability to create a single-use and/or reusable Course Set |
| Ability to create Course Set that contains proposed or approved courses |
| Ability to create a Course Set that contains courses and/or other Course Sets |
| **Manage Reusable course set** |
| Ability to modify reusable Course Sets |
| Ability to retire reusable Course Sets |
| Ability to delete reusable Course Sets |
| Ability to version reusable Course Sets |
| **Learning Objectives** |
| **Define learning objectives** |
| Ability to author Learning Objectives |
| Ability to edit Learning Objectives |
| Ability to delete Learning Objectives |
| Ability to search for, copy and then edit Learning Objectives |
| **Organize Learning Objectives** |
| Ability to order Learning Objectives |
| Ability to create a hierarchy of Learning Objectives with levels and sublevels |
| Ability to create Learning Objectives categories |
| Ability to maintain an inventory of Learning Objectives categories |
| Ability to manage an inventory of Governed Learning Objectives |
| Ability to categorize Learning Objectives |
| **Attach Learning Objectives** |
| Ability to attach Learning Objectives to a Course |
| Ability to attach Learning Objectives to a Course Activity |
| Ability to attach Learning Objectives to a Program |
| Ability to map Program Learning Objectives to Courses that support them |
| **Course Analysis** |
| **Ability to determine the Dependence of Other Courses on a Selected Course** |
| Ability to determine which Courses are Jointly-Offered with the Selected Course |
| Ability to determine which Courses are Cross-listed with the Selected Course |
| Ability to determine which Courses use the Selected Course in a Prerequisite  Rule |
| Ability to determine which Courses list the Selected Course in a Co-Requisite Rule |
| Ability to determine which Courses list the Selected Course in a Anti-Requisite Rule |
| Ability to determine which Courses list the Selected Course as a Recommended Preparation |
| Ability to determine Oversight Organization(s) of the Dependent Courses |
| **Ability to determine the Dependence of Programs on a Selected Course** |
| Ability to determine which Programs list the Selected Course as an Entrance Requirement |
| Ability to determine which Programs list the Selected Course as a Satisfactory Progress Requirement |
| Ability to determine which Programs list the Selected Course as a Completion Requirement |
| Ability to determine Oversight Organization(s) of the Dependent Programs |
| **Ability to determine the membership of a Selected Course in Reusable Course Sets** |
| Ability to determine which Reusable Fixed Course Sets contain the Selected Course |
| Ability to determine which Reusable Dynamic Course Sets contain the Selected Course |
| Ability to determine Oversight Organization(s) of the Dependent Course Sets |
| **Ability to access details of the Dependencies** |
| Ability to filter the analysis results |
| Ability to view the Dependent Course from the analysis results |
| Ability to view the Dependent Program from the analysis results |
| Ability to view the Dependent Course Set from the analysis results |
| **Program Analysis** |
| Ability to view which Courses are shared across two or more Selected Programs |
| Ability to run an analysis of a Select Program based on Learning Objectives |
| Ability to determine the Prerequisite chains for required Courses in a Selected Program |
| Ability to view which Courses associated with the Selected Program are not owned by the same Oversight Organization as the Selected Program |
| **Other** |
| Ability to print the results of a Dependency Analysis |
| Ability to export the results of a Dependency Analysis |
| Ability to automatically trigger a Dependency Analysis as part of a Modify or Retire Course Proposal |
| Ability to send notifications to Oversight Organization(s) of Dependent Courses, Course Sets, and Programs |
| **Backend Services** |
| Academic Time Period Service |
| Canonical Learning Unit Service |
| Comment service |
| Course Service |
| Document service |
| Enumeration Management Service |
| Learning Objective Service |
| Dictionary Service |
| Learning Result Catalog Service |
| Message Service |
| Organization Service |
| Program Service |
| Proposal service |
| Search Service |
| State Service |
| Statement Service |
| Type Service |
| Version Management Service |
| **Configurations** |
| Institutions |
| User interface |
| Themes |
| Layout, Views, sections & fields |
| Messages, labels & help text |
| Service dictionaries |
| Searches, pickers and browsers (e.g. Catalog browser) |
| Rules, Enumerations, field bindings |
| Business processes and workflows |
| **Accreditation** |
| Accreditation application |
| Results of initial or preliminary visit |
| Approval results of accreditation commission |
| Address initial visit recommendation |
| Self-study results |
| Full visit result details |
| Visiting committee report |
| visiting committee recommendation accreditation status |
| Commissions actions on status or conditions |
| Refinement and submission of schoolwide action plan |
| Submission of interim progress reports |
| Subsequent reviews and reaffirmation of accreditation |
| Systematic improvement and renewal |
| **Misc.** |
| Single Sign on |
| API Integrations |
| Bilingual Site |
|  |
| **System features** |
| Authentication & authorization |
| Auditing & logging |
| Exception handling |
| Manage files and directories |

# APPLICATION ARCHITECHURE

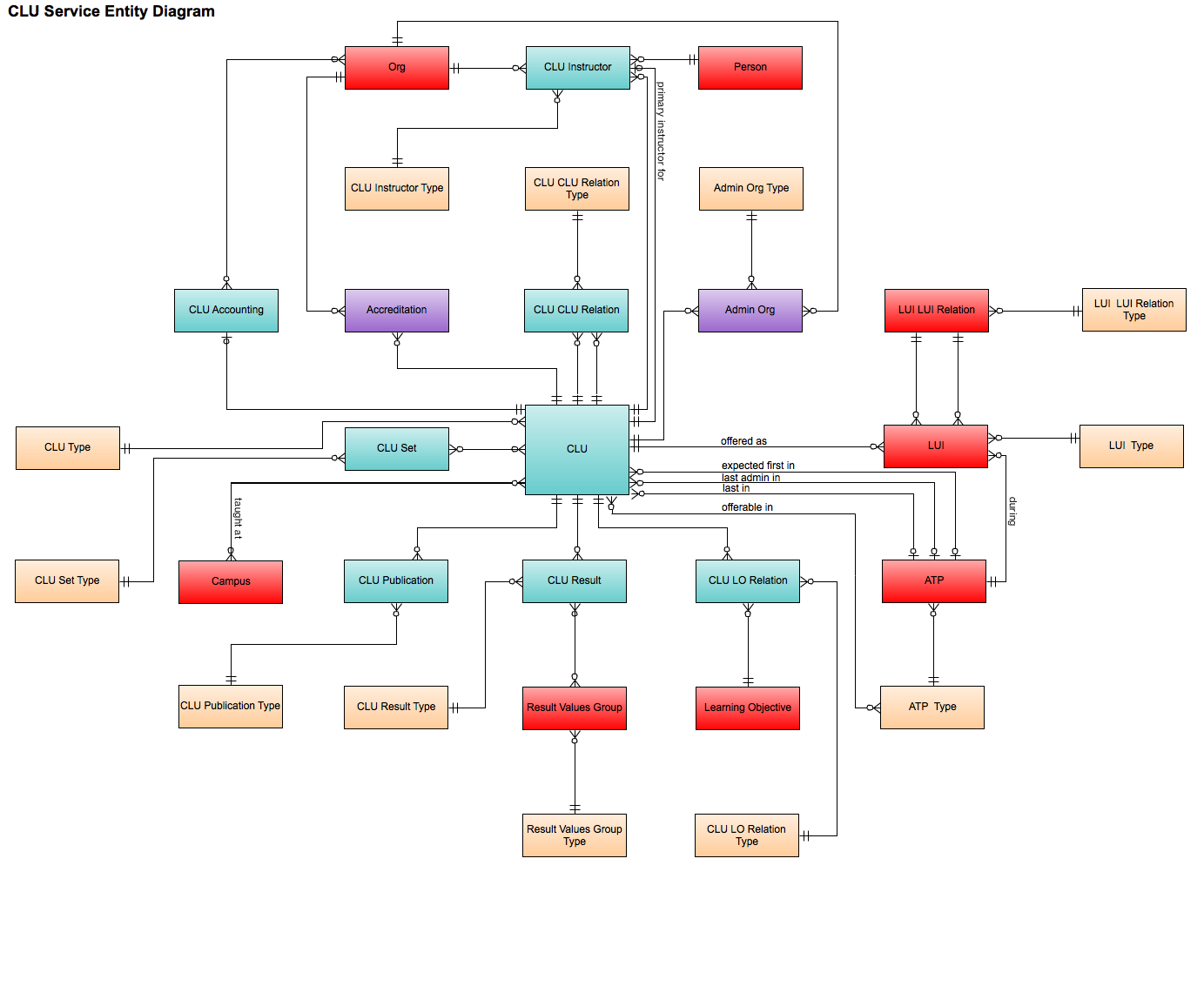
## COURSE CREATION WORKFLOW



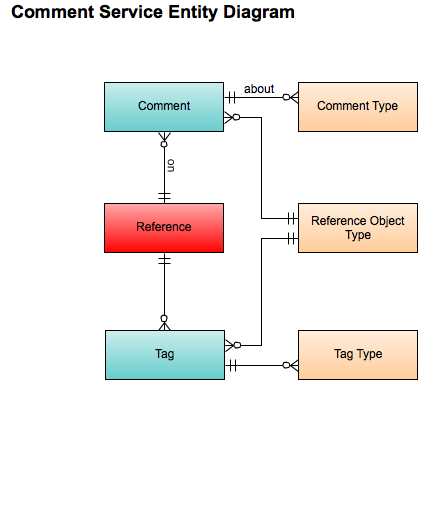
## ENTITY RELATIONSHIP DIAGRAMS

### ATP Service

### Canonical Learning Unit



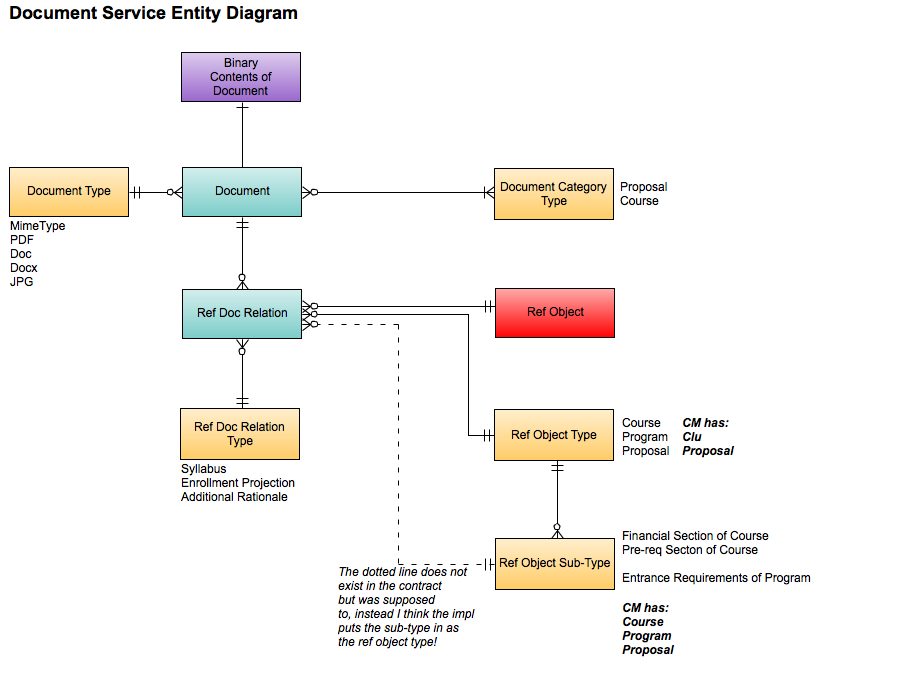
### Comment Service



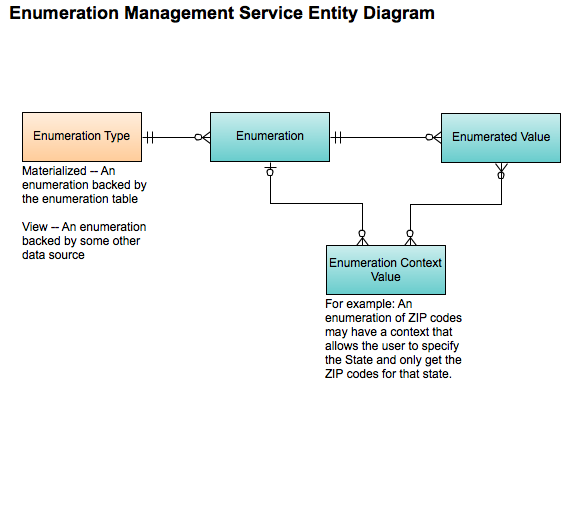
### Course Service



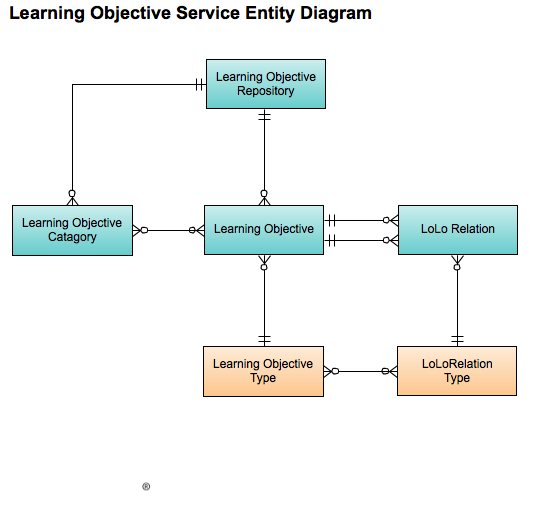
### Document Service



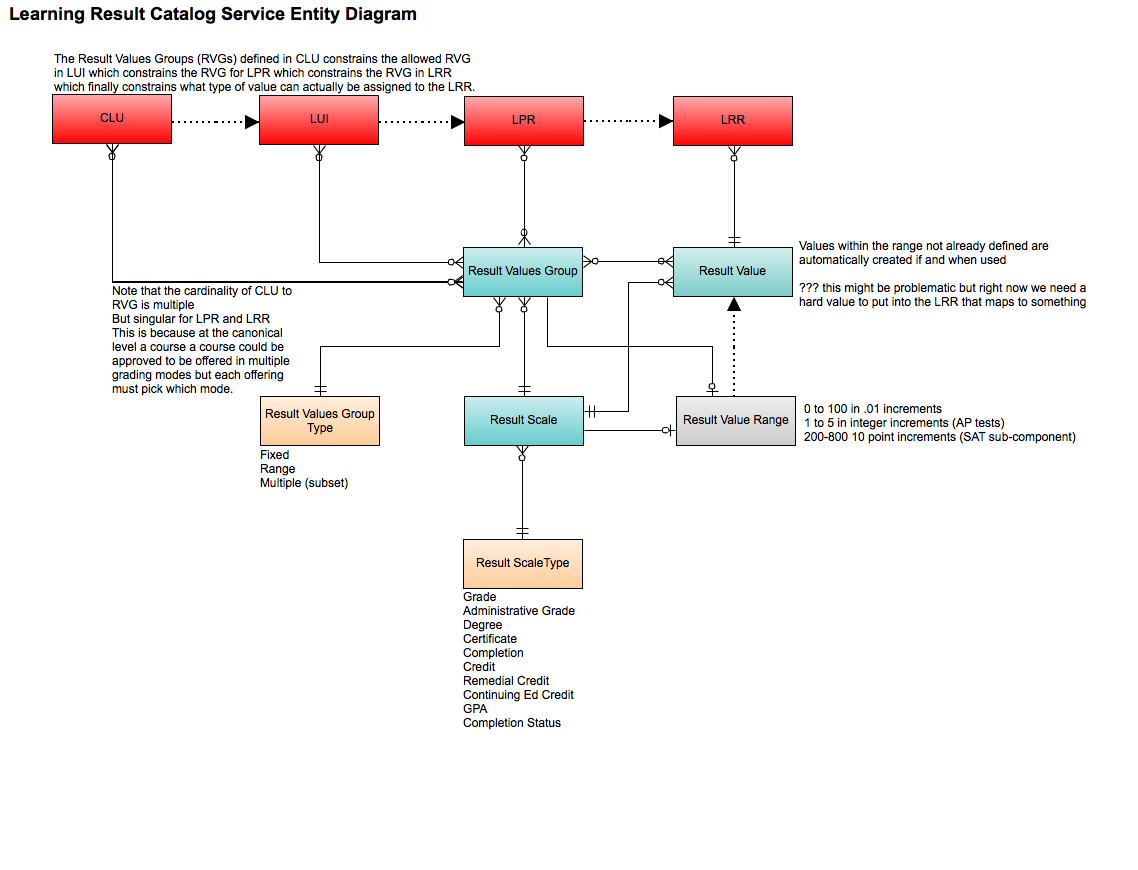
### Enumeration Management Service



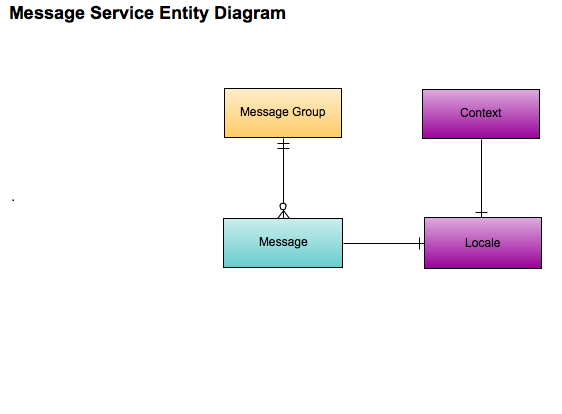
### Learning Objective Service



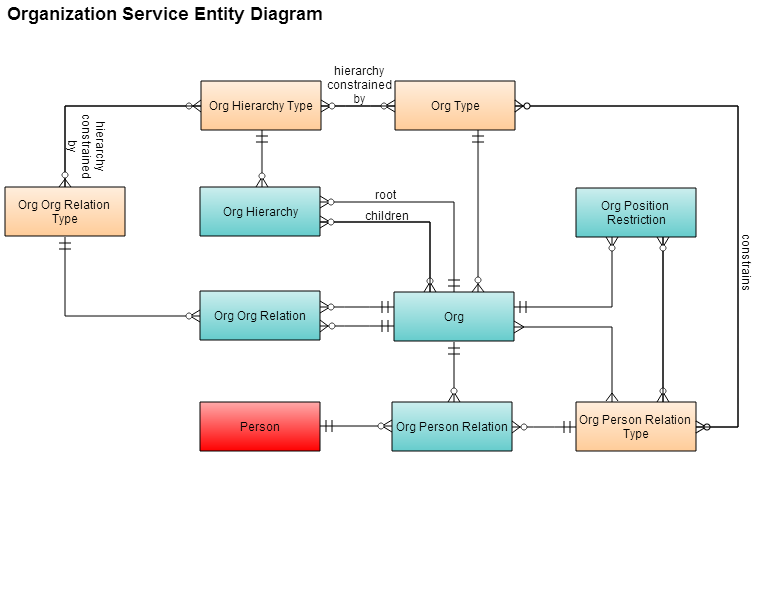
### Learning Result Catalog Service



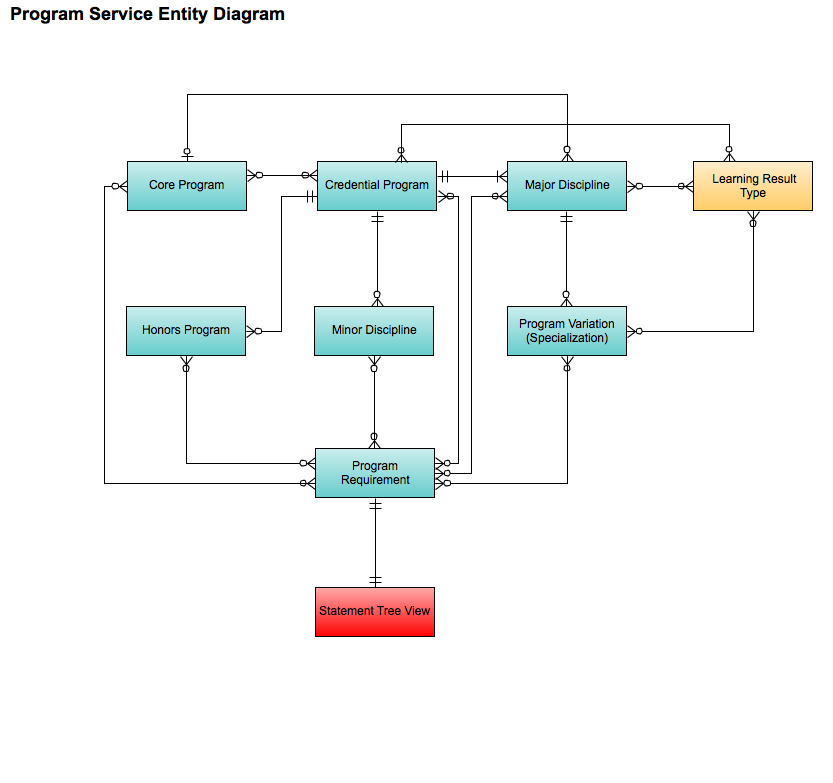
### Message Service



### Organization Service

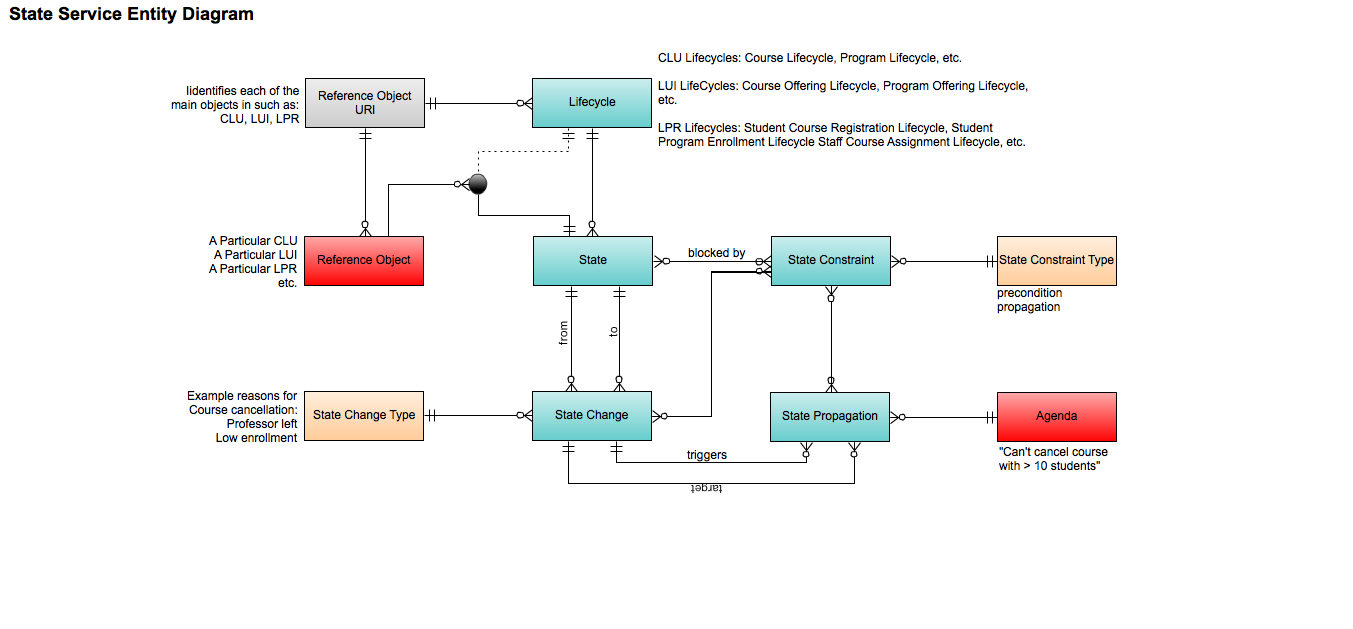


### Program Service

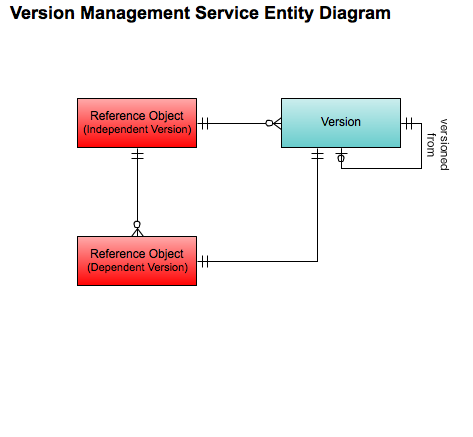


### Proposal Service

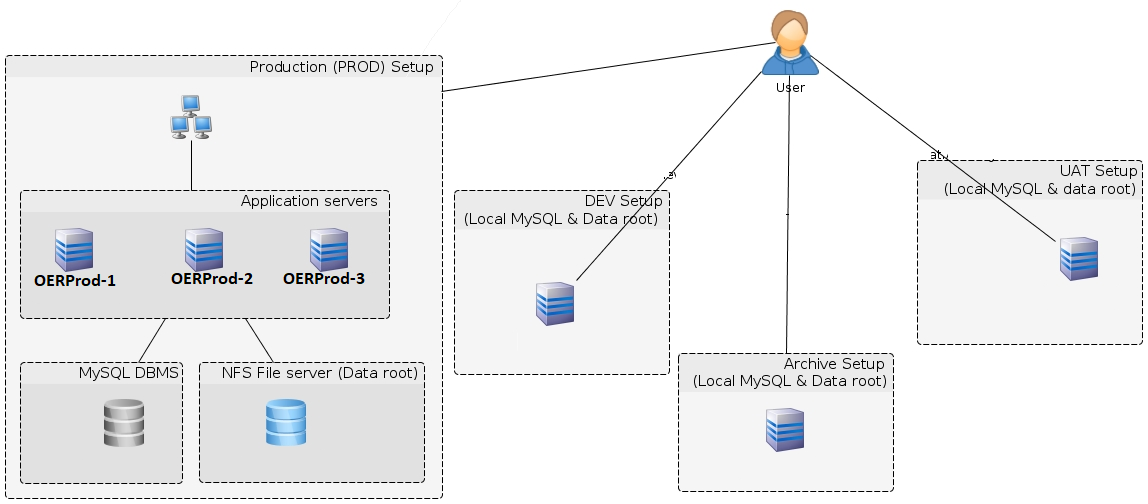
### State Service



### Version Management Service

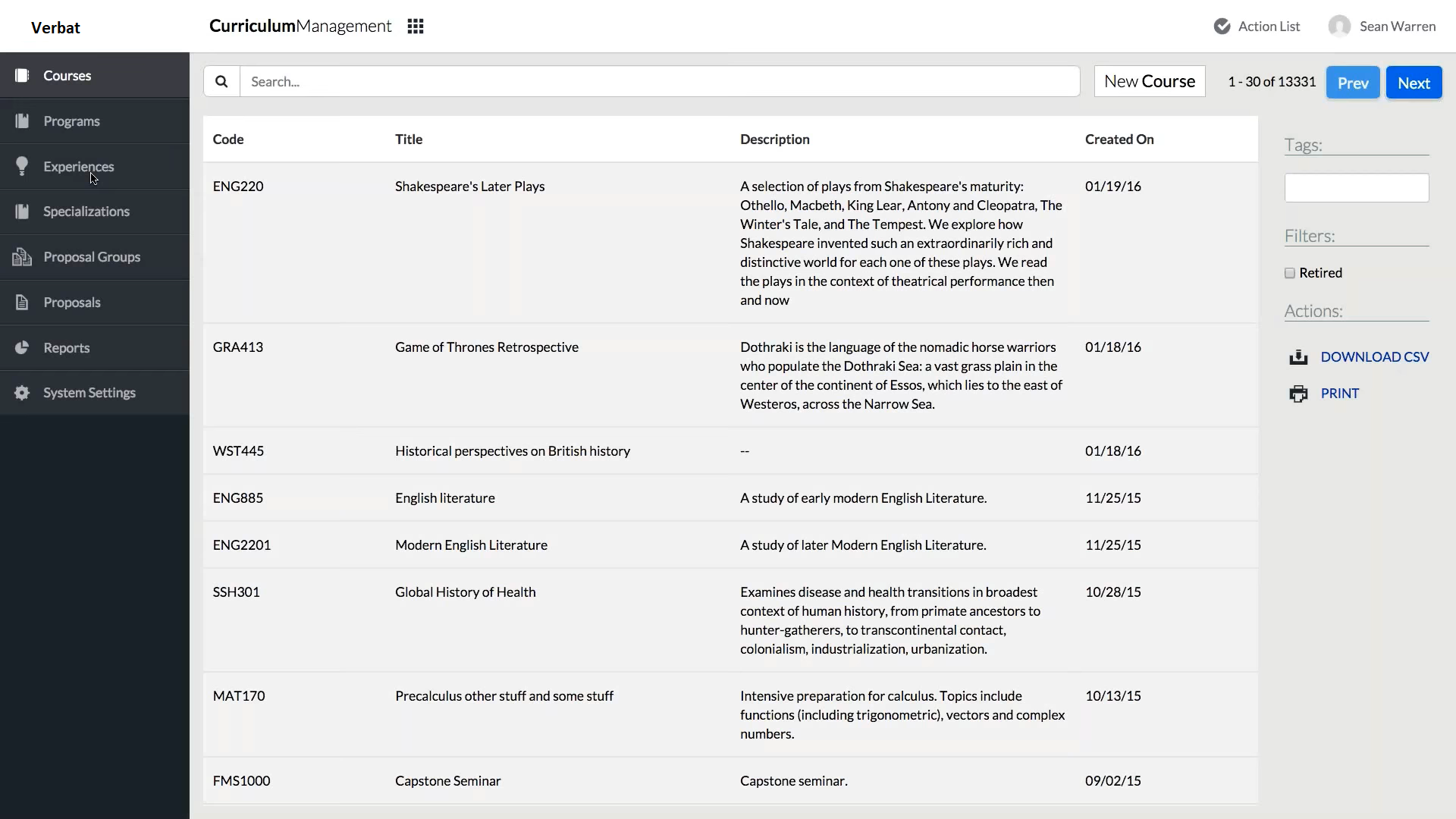


## LOCAL SERVER INSTANCE (On Premise)

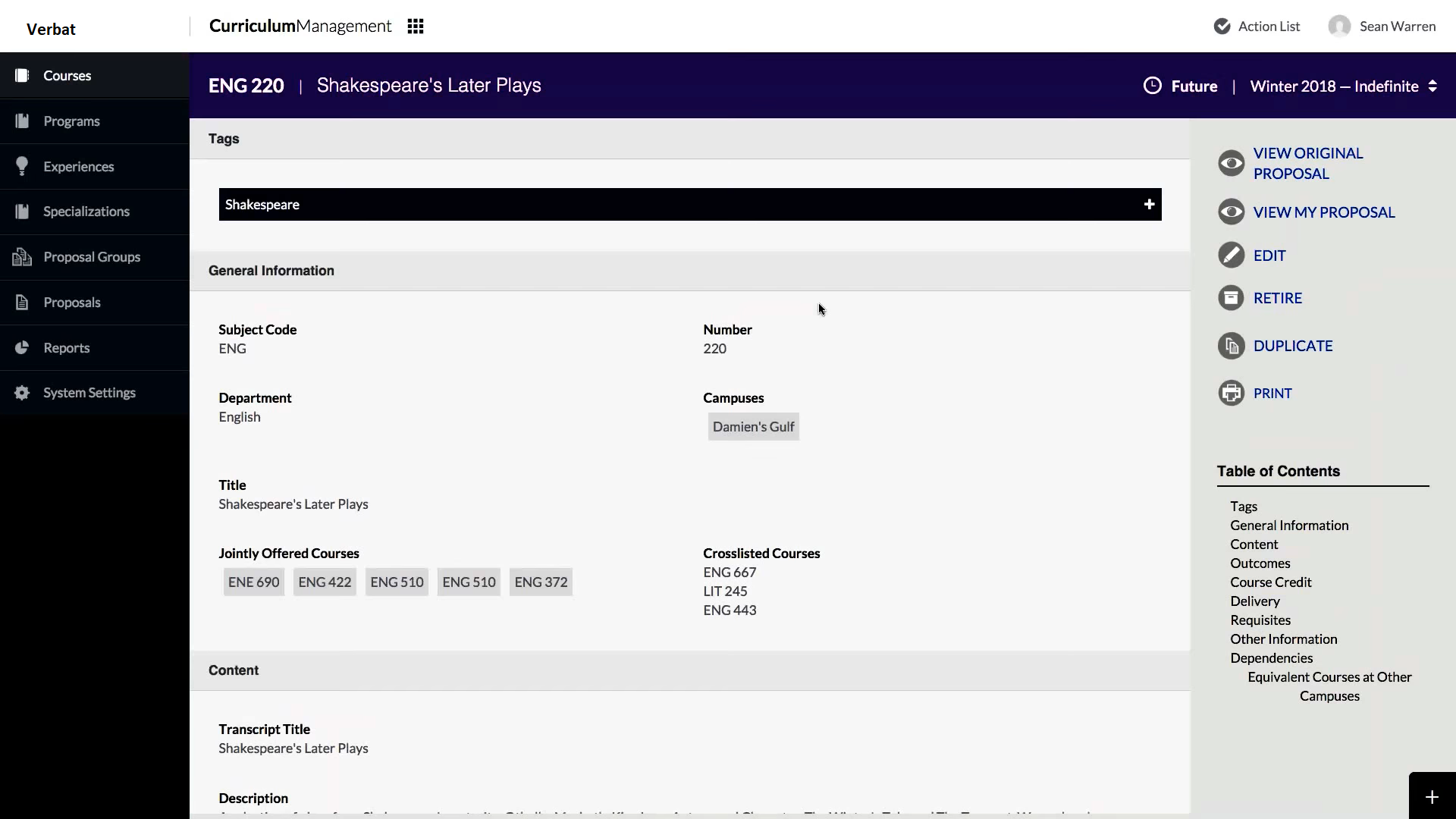


## Sample Screens

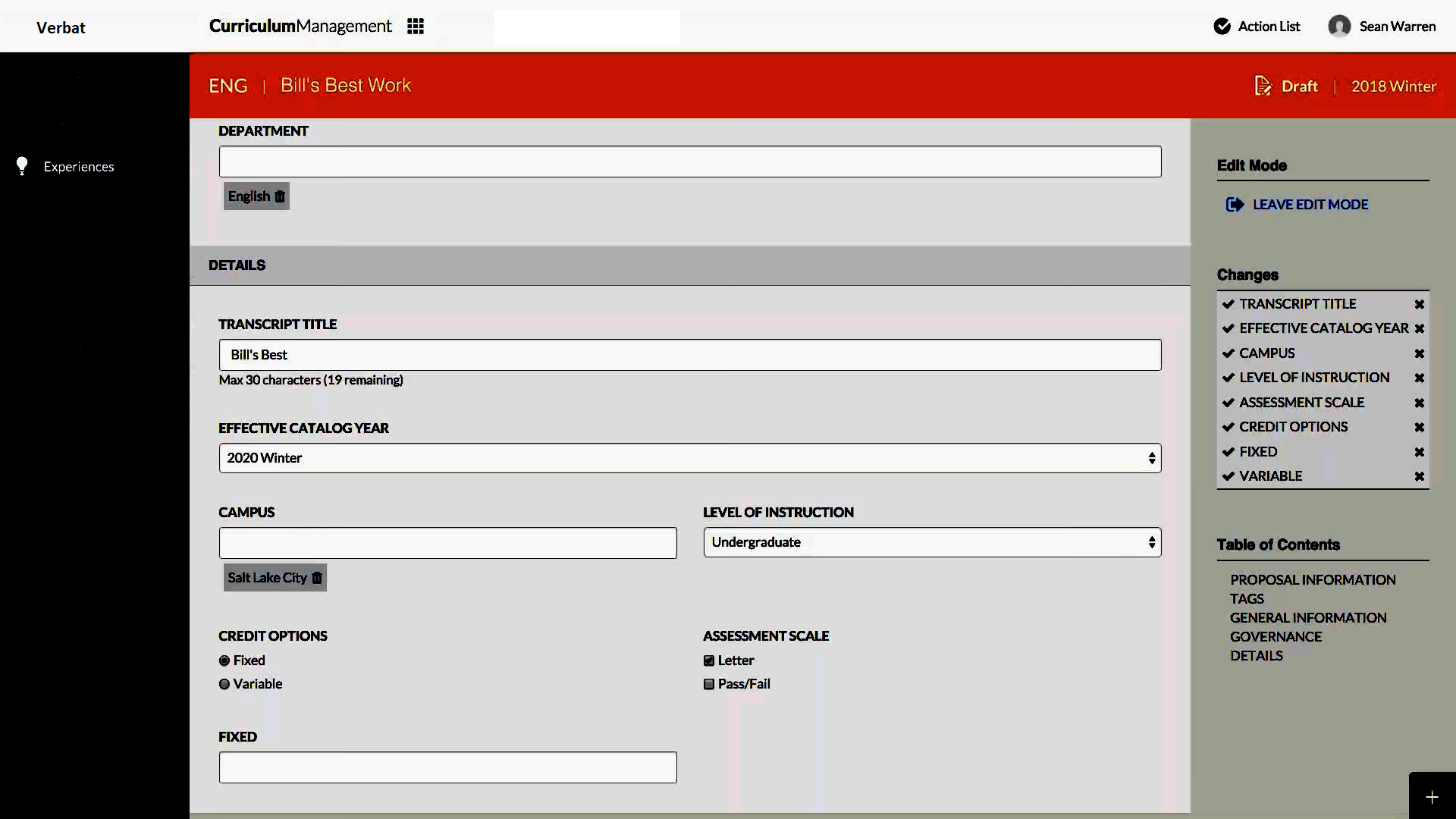
### Course List



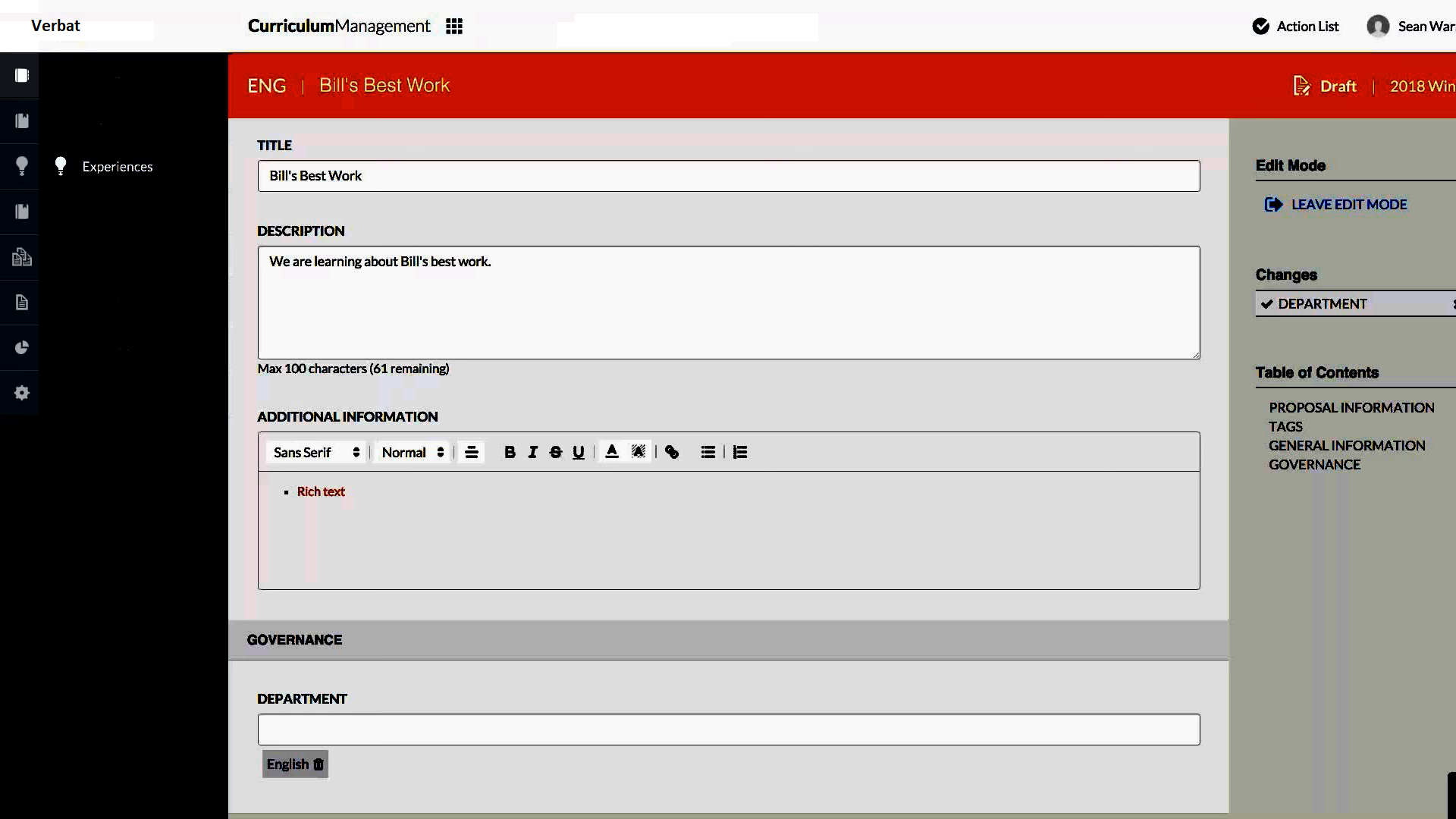
### Course View



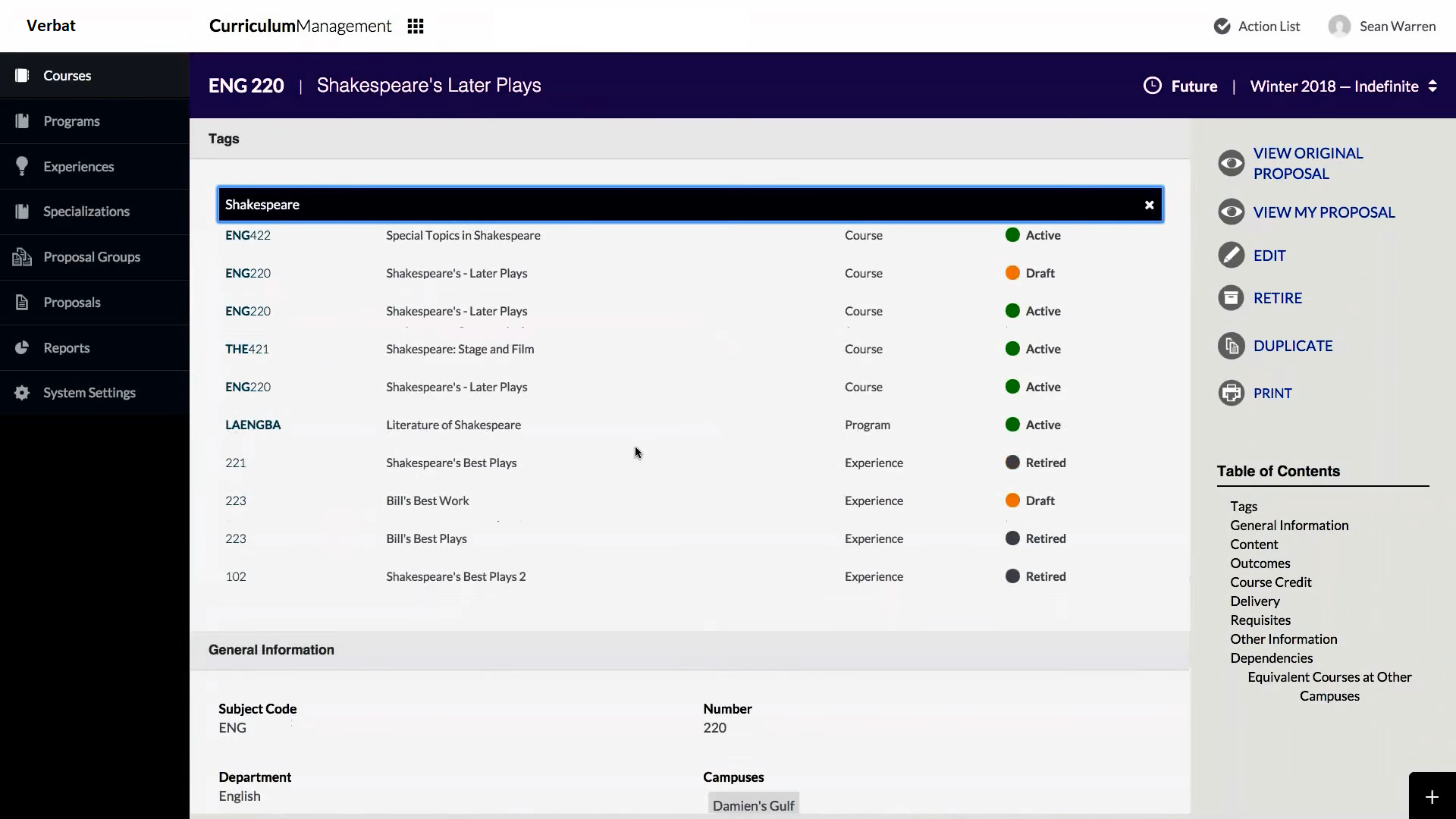
### Form Changes (Tracking)



### Rich Text Content



### Tagging Courses



## NON-FUNCTIONAL REQUIREMENTS (OTHERS)

|  |  |
| --- | --- |
| **Requirements** | **Details** |
| User Experience and  UI Design | * The application will be developed only in English * The layout and graphical components will be created considering the usability factors |
| Performance | * Application will allow users to have smooth and quick access to the information or services they require. |
| Security | * Web security standards will be followed. |

## TECHNICAL CONFIGURATIONS

### DEVELOPMENT ENVIRONMENT

* Java, Maven, Spring, SVN, TOMCAT
* MySQL or Oracle, SoapUI, IMPEX
* HTML / CSS 3
* PHP, Code ignitor
* Eclipse or IntelliJ, JDK

### RECOMMENDED WED HOSTING PACKAGE - DEDICATED

* Operating System: Windows Server
* CPU: 2 cores
* Domains: Unlimited
* Disk Space: 200 GB
* Monthly Bandwidth: 50 GB
* Apache Tomcat
* JavaVM, SDK
* Hosting will be on cloud according to the band-width.

### BROWSER

The application developed shall be compatible with the browsers listed below:

* Internet Explorer 11
* Mozilla Firefox 50 or above
* Chrome 50 or above

### HARDWARE DEVICES

#### Mobile device with the following specs

* The device should support LTE / Wi-Fi 802.11 a/b/g/n/ac
* The device should have 3 GB RAM and above memory

#### Web Server

We recommend the specification mentioned below for the best output:

* Microsoft Windows Server 2016 with IIS 7.5 +
* Processor: 4.20 GHz Intel Core i7-7700 or equivalent
* Memory: 16GB
* Disk Space: 1 TB of free disk space

### TECHNICAL STANDARDS

* Testing Devices

Google Pixel     Android 7

Motorola Moto G Turbo Edition -   Android 6

* OS Version

Android: Android KitKat and above

IOS: IOS 9+

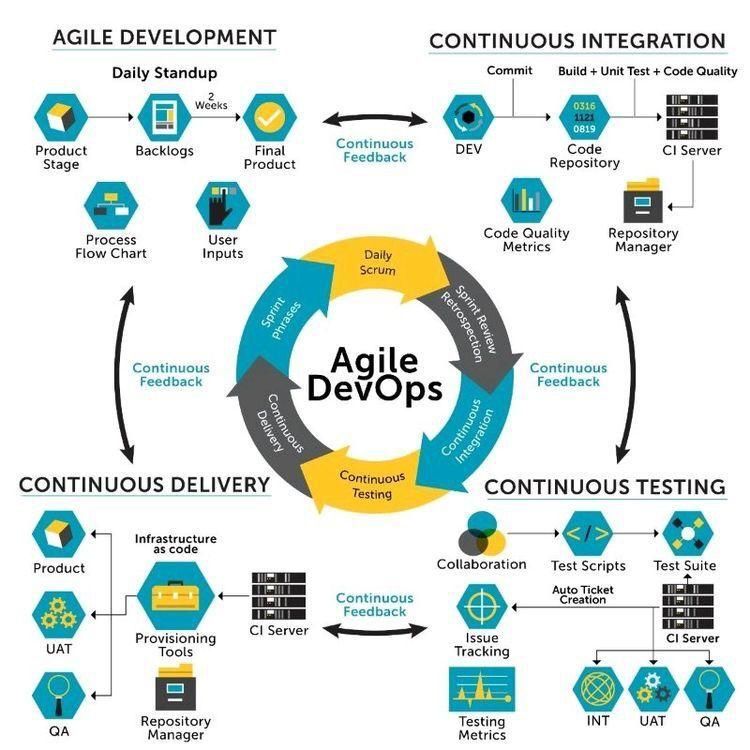
### TECHNICAL GUIDELINES

These guidelines provide instructions and conditions that will be adhered to during the development of the mobile application.

* API will be used, as the case may be, in realizing the features and functionalities mentioned.
* The client will finalize the functional requirements and UI/UX before the commencement of the project.
* Verbat will be testing the app in the mentioned devices only. Testing on devices other than the ones mentioned under the “Technical Standards “will have to be specified and provided by the client at the beginning of the development phase.
* The client will have to provide the details of the testing devices they are using before the start of development phase.
* Client should provide the relevant Developer's Account credentials before the development phase. In case Verbat needs to create the developer id, additional charges will be incurred by the client.
* The duration mentioned in the project timeline is for the development and testing; any delay or time taken by the review team to respond will not be Verbat’s responsibility.
* Any clarification required from client needs to be addressed within 02 business days.
* The apps will be developed / created within the guidelines of respective play store.
* Verbat will strictly follow the guidelines provided by the respective stores.
* Verbat will inform the client if any of the client requirements / request deviate from it.
* Customization of the features of the app will be susceptible to the limitation imposed by the respective platform/ store.
* Once development commences, the test device/screen sizes will not be susceptible to change. Any change requested by the client will have to go through change management.
* OS version support will be limited to the ones mentioned in the technical specifications. Further support will have to go through change management.

## PROJECT DELIVERY

### Agile Methodology



Verbat’s philosophy has been to deliver products that allowed clients to be intimately involved with the development activity. As early as 2005 Verbat had adopted the agile development practices into its pipeline for software development. This allowed clients to closely observe the product being developed. It created a tight feedback loop that allowed us, along with our clients to better understand the requirements and build a right sized and proportional product.

With the advent of next generation applications and tools that support a wide range of activities related to development, deployment and integration; the feedback loops became tighter and coupled, while at the same time it allowed us to create applications that were decoupled. Thus giving us the capability to have greater flexibility in development and deployment.

While development broke down the barrier between developers and management, Devops broke the barrier between software developers and operations teams. Our development and operations teams work under a single silo. While our Scrum teams are cross functional, in our quest to adopt better integration with operations, we have realized that Devops is not just a set of tools and processes, but a mindset and culture. We have fostered a culture collaboration and communication. Our scrum teams and Devops teams share a symbiotic relationship and they work united towards a common goal.

As a company, we have adopted the following tenets in our Devops philosophy

**Speed:** i.e. Move at a high velocity so as to innovate faster for customers and adapt to changing markets better, and grow more efficient at driving business results.

**Rapid Delivery:** Increase the frequency and pace of releases so we can innovate and improve products faster and thus respond to customers’ needs and build competitive advantage.

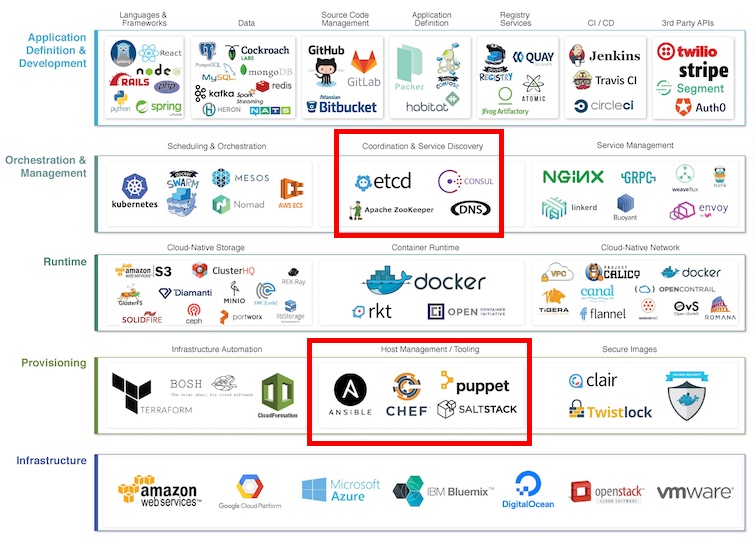
**Reliability:** Ensure the quality of application updates and infrastructure changes so we can reliably deliver at a more rapid pace while maintaining a positive experience for end users.

**Scale:** Operate and manage infrastructure and development processes at scale.

**Improved Collaboration:** Build more effective teams under a DevOps cultural model, which emphasizes values such as ownership and accountability. Developers and operations teams collaborate closely, share many responsibilities, and combine their workflows.

**Security:** Move quickly while retaining control and preserving compliance. We adopt a DevOps model without sacrificing security by using automated compliance policies, fine-grained controls, and configuration management techniques.

Please see below an illustration of our technology stack. The tools of the trade that allow us to practice an Agile and Devops centric framework



### PROJECT MANAGEMENT

The Verbat development center strictly follows industry standards on quality. Our project management process is governed by the Verbat Quality Management system and is verified through internal audit programs.

Verbat will dedicate a project leader for the proposed implementation. Furthermore, Verbat proposes the client to identify one project manager (*CSPM – Client-Side Project Manager)* who will be driving all activities undertaken by the client, and will be the single point of contact for Verbat.

### ROLES & RESPONSIBILITIES

Verbat will assign its own dedicated Project Lead (*VPL – Verbat Project Lead*) for the client. The VPL will be responsible for planning and management of all activities related to the project. Furthermore, the VPL will work closely with CSPM, on all periodic status updates and will ensure high level visibility and comfort on the progress of the project.

### DELIVERY ACTIVITY SUMMARY

|  |  |
| --- | --- |
| Activities | Description |
| Proof of Concept | Working Model of the project with two languages. |
| Detailed Requirement Analysis | Verbat team to conduct detailed study of requirement for the phase. If clarification is required, the team will reach out to Client for more information and/or time for discussions. |
| DB Design | DB design for central and test DB. |
| Software Requirement Specification document (SRS) | Once the requirement analysis is completed, Verbat team will submit the SRS document for approval |
| UI/UX Design, Prototyping | Based on the SRS, Verbat UX/UI team will work on the UI/UX of the screens and submit prototype for approval |
| Functional Specification Document (FS) | Once the UI/UX is approved, Verbat shall submit an FS document for approval. |
| Development | Actual system development starts based on the FS. This involves detailed design and software development of Web Application. |
| Testing | Test Planning, test plan creations, internal, integration testing and user acceptance testing. |
| Deployment | Deploying the latest build in the Verbat Test Server. |

### PROJECT IMPLEMENTATION PLAN

Verbat will be providing the solution in a stand-alone fixed bid approach which ensures minimum viable solution for quick wins with core focus on the long-term business objective and outcome. Once the implementation is over, Verbat will initiate the application maintenance process (*once the maintenance contract is signed*) which continues to extend after the implementation.

### DELIVERABLES

* Proof of Concept
* Project Plan
* Software Requirement Specification Document (SRS)
* Prototype Design
* Functional Specification Document (FS)
* Fully Developed & Tested Application
* Source Code

### ESTIMATED DELIVERY TIME

The timeline estimated for delivering the application will be as below:

* Working days to complete the proof of concept on completion of resource mobilization.
* Working days to complete the SRS & prototype of Language Translation Application.
* Working days to complete the development.

| **Activity** |
| --- |
| Contract Signoff (T0) |
| Proof of Concept-Allocate Resource |
| Proof of Concept-Complete |
| Proof of Concept- Approval (T1) |
| Detailed Requirement Gathering |
| Software Requirement Specification Document (SRS) |
| SRS Approval (T2) |
| System UI/UX Prototype-Complete |
| System UI/UX Prototype-Approval (T3) |
| Functional Specification (FS) |
| FS Approval (T4) |
| Development Phase-Complete |
| Perform QC (Unit Testing and Integration Testing) |
| System ready for UAT |
| UAT Acceptance on Verbat production server (T5) |

*Note:*

* *The above-mentioned timeline is in Working Days.*
* *Upon project confirmation, Verbat requires a lead time of minimum () working days for resource mobilization for proof of concept.*
* *Resource mobilization will be initiated post the confirmation of the project along with LPO, signed*

*proposal and advance payment.*

* *The initiation of the UI/UX/Prototype development is dependent on the confirmation of SRS.*
* *The above-mentioned timeline for development is post-confirmation of FS.*
* *Documentation submitted after project initiation and system study supersedes any proposal or documentation submitted during initial requirement gathering / discussion / negotiation.*
* *Project plan will be submitted post the confirmation of project with necessary payments.*
* *Any delay in getting the approvals of deliverables from client will cause change in timelines and the revised timelines will be updated in weekly status reports shared with client after the project commencement.*
* *All approvals and queries regarding the client requirement and any queries which may hinder the project advancement at any stage should be answered by the client within () business days from the time of initiation, failing which, the time delay will get added to the actual effort and timeline estimated.*

### DEPLOYMENT DETAILS

* Cloud server for hosting the application will be the responsibility of the client.
* Verbat will deploy the application on client’s cloud server after the client conducts the necessary acceptance testing.

*Note:*

* *Hosting the application at Verbat’s production server (hosting charges) will call in for additional charges. In the event, client wants to procure SSL, the same can be provided by Verbat at additional cost.*
* *Only deployment of the application will be the responsibility of Verbat, any additional installation will be charged separately.*

### RELEASE PLANNING

* Client will be informed about the release date and time through email.
* Client performs the UAT

### RISK CONTINGENCY PLANNING

Verbat has identified various risk factors associated with this assignment and understands the impact of these risk factors on the project schedules. The objective of this section is to highlight for both Verbat and client, the risk factors, to analyze the impact of the risks on project execution, and to propose strategies to control and reduce the impact of the risk factors. These various risks, which could arise during the project, are tabulated below along with mitigation implementation.

| **Type of risk** | **IMPACT** | Risk Mitigation | Risk Handling |
| --- | --- | --- | --- |
| Scope Creep | **H** | Functions and features will be detailed in system requirement document and will go through client approval. Once this document is approved, all changes will go through the change management process for impact assessment. | Proper change management procedure will be implemented |
| Delay in client feedback | **H** | The plan is prepared with sufficient lead-time for reviews and approvals.  The client will advise us on all dates connected to the document review and approval. | The request for feedback will be escalated if not attended to at the right time so that the schedules are not affected. Deemed acceptance criterion is finalized up-front and will be followed |
| Non-availability of necessary software, frameworks, database instances and infrastructure at client’s hosting environment (If hosting support is provided by Verbat) | **M** | Client will be informed in advance on these requirements. | Possible impact to schedule |
| Manpower attrition | **L** | All efforts will be made to ensure that all initiatives are process dependent. To mitigate risk Verbat/Client will train a person to ensure all back-ups are in place. | A new person will be appointed as early as possible, provided the required project-specific training and mentoring is in place - to minimise impact of attrition on the project |

*H-High, M-Medium, L-Low, NA-Not Applicable*

## PROJECT ASSUMPTIONS

The project solution and technology are created from the initial understanding of the requirement shared with Verbat through mails and meetings. The proposed solution is based on the following assumptions:

### OBJECTIVES

* The requirement is to develop a Language Translation Application (web, android, iOS) with the functionalities as defined in ‘Functional Requirements’ section

### DESIGN

* Client to provide Verbat with the brand guidelines.
* Color theme shall be provided by the client.
* Client shall provide licensed images and logos in specified size & format.
* Client shall provide the text and associated images for the proposed application. Text should be provided in digital format preferably in MS Word 2013 or above
* Verbat is free to use custom-made template for design, if required.
* The application designed for mobiles will be in portrait mode

### DEVELOPMENT

1

Client shall procure the following services:

* Mesibo Services
* Google Transcribing Services.
* Google Translation Services.
* Application shall be developed as a hybrid application so that it can be deployed on both IOS and Android.
* Additional charges may be incurred for archiving and encryption services (depending on the requirements).
* Peers making use of translation services are likely to experience a latency which is directly proportional to the bandwidth of their network connection.
* Peers making use of translation services may have to converse through an intercom like interface.
* The application requires a fast internet connection for real time communication.

# **OUT OF SCOPE**

With the ever-evolving digital market, the requirement should be clear to both the parties involved, hence the importance of mentioning the out of scope details of the project. Following are considered to be out of scope while creating this proposal:

* Purchase of images, fonts
* Any language other than English
* Migration of existing data / Database migration
* Content writing / proof reading / Data Replication / Manual data entry
* Content or image procurement or uploading or editing
* Native iOS App Development / Android App development
* Developer account creation and Maintenance (*IOS and Android Store*)
* Adding new features to the application other than mentioned in the functional specifications. Such requests will be handled via change management. For Change management details, please refer to section titled “Change Management” in the Proposal. (*refer page no: 33*)
* Annual Maintenance Contract (*Bug fixing, debugging, enhancements*) – Please refer to section titled “Maintenance and Support”, unless contracted for. (*refer page no: 34*)
* Hosting Infrastructure and Maintenance (*web and email hosting*), unless contracted for.
* Backup solution and Disaster recovery unless contracted for.
* Physical deployment on-site / installation of the application in devices and physical connection, installation of system.
* Integration with third-party, if any, other than mentioned in the functional specifications
* Hardware Integrations / procurement and purchase
* Procurement of SMS gateway / payment gateway / email gateway
* Integration of SMS gateway / payment gateway
* SSL Purchase and installation, if any
* Plugin/template purchases, if any
* API purchases
* OS other than mentioned in the Hardware Interface
* Relevant / related software libraries
* Mobile offline access or operations

# **CHANGE MANAGEMENT**

Any addition which comes out of the project scope, upon and after the launch of the website will be considered as change management. Verbat recommends the following change management procedures for the same:

* Changes will be implemented only after raising a change request.
* Change requests will be studied and an impact analysis will be performed on the existing work flow.
* Upon assessment of the impact, effort estimation will be calculated and raised as an additional requirement.
* The change request will be initiated only after receiving a formal approval from the client for the additional changes raised.
* Changes which are out of scope will be charged @ per man day rate.

Activities for change scheduled

Modules checked in

Unit testing done

Change request accepted

Informed of the Action

Implementation of changes

Modules checked out

Activities planned and scheduled

Effected modules identified

New system released with change request

Changes made & documented

System

Need for change recognized

User submits change request

Evaluate for cost, schedule & effort

Change control authority

yes

No

## MAINTENANCE & SUPPORT

* Maintenance contracts by default are supported as per the basic SLA terms.
* AMC with Basic SLA is charged at % of the total project value. Additional Effort/change management request will be added towards Total Value of the Project to determine the AMC value.
* Maintenance support is limited to providing application support for ensuring the consistency of the look-and-feel, bug fixes and user issues i.e. maintenance and support of the existing features of the application.
* Support does not in any way cover providing technical or other support to the end users or hardware support. The maintenance agreement does not include functionality changes or feature additions which are handled as change requests which will be charged per man day rate. AMC does not include server support, maintenance and application deployment.
* AMC charges will cover Off-Site Support and Debugging. Support includes E-mail, telephone and video conference (if required). In the event, the application is hosted with the client or if it is a client server development; necessary remote desktop connectivity should be provided for carrying out maintenance activities.
* All maintenance support will be executed by Verbat off-site team. In the event of any need for on-site work, all expenses incurred for such trips (flights, accommodation, meals transportation) will be payable to Verbat by client.
* Gap in AMC - In case if the client does not opt an AMC for a year and wants to renew it after that period, % of the AMC amount for the year for which AMC is not taken will also be payable if the client wishes to renew the AMC contract.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SLA Type** | **Max Response Time** | | **Max Resolution Time** | **Target** |
| Basic | 1 working day | 3 working days | | Request / incident / problem tickets |

*Note:*

* *Time zone applicable as per India Time Zones (3:30 GMT to 12:30 GMT, Monday to Friday)*
* *Please note that the AMC support shall start only after all the necessary sign-offs (AMC Document) to this effect have been given.*
* *It is not mandatory that the client should opt for an AMC. The client will still be supported on an ad-hoc basis on an agreed man-day rate.*
* *AMC Payment Terms: 100% to be paid as advance.*

AMC Option: Client can opt for time and material based Annual Maintenance, the details of which will be shared post the completion of project.

# **TERMS AND CONDITIONS**

## ACCEPTANCE CRITERIA

* UAT (User Acceptance Test) sign off should take place within Days from the first release of the application and the acceptance confirmation needs to be mailed to Verbat failing which Verbat will consider the project as approved by the client.
* Any comments and reasons for rejection need to be documented and the same needs to be sent as an email from the official e-mail id of client to Verbat on or before days from the first release.
* Timeframe for acceptance for any further release will be mutually agreed and finalized between client and Verbat depending on the UAT Comments

## WARRANTY

* Verbat shall provide a bug ﬁx warranty at no additional cost for 30 days from the date of acceptance of the project, for correction of any errors in the developed application that may be attributed to Verbat.
* However, this does not cover modifications by Client, or use of the application on an environment other than the proposed environment, or other circumstances outside Verbat’s reasonable control. In such cases Verbat reserves the right to charge for its services.
* All error corrections will be executed by Verbat offsite team. In the event of any need for on-site work, all expenses incurred for such trips (flights, accommodation, meals, transportation) will be payable to Verbat by Client.

## SOURCE CODE & INTELLECTUAL PROPERTY RIGHTS

* Upon completion of the Project and 100% completion of the payment, the client will have access to the source code except for proprietary codes, developer tools and third-party applications etc.
* The solution offered will be the intellectual property of the client and will be made available to the client on an “unlimited license” basis.
* No person or organization, other than Verbat or any person authorized by Verbat in writing, has permission to modify/change the software solution to be eligible to get continued support from Verbat as per the support terms defined under this document.
* Verbat accepts no liability or damages of any kind arising out of use or non-use of the software delivered. The responsibility of testing of software lies with the client.

## GENERAL TERMS AND CONDITIONS

* Offer Valid for 30 calendar days from the date of submission of the Proposal.
* An average of 20 working days are assumed in a month.
* All the development activities will be carried out from our off-shore development center in India.
* All the documentations will be provided in English.
* Third party components may be used to develop this application.
* The scope of the project is to develop the application as detailed in the scope of the project and mentioned in this proposal. Any changes or additions will have to go through our change management process.
* This proposal is derived and concluded from either the RFQ /RFP/data shared via email / information transferred during an initial requirement analysis meeting / tele-conversation. Verbat reserves the right to change the terms of this proposal if the final terms (including the costing), features & functionalities and timelines are changed during the course of the project. Hence any fees quoted / timeline committed in this proposal may not be considered as final unless agreed upon and signed by both parties.
* Web Application will be best viewed only in the environment mentioned in the section Browser Compatibility.
* Mobile app will be best viewed only in the environment mentioned in the section Hardware Interface.
* All source codes and other project artifacts will adhere to the Verbat document templates and internal coding standards.
* The documents delivered to the client include the ones mentioned under ‘Deliverables’ and these will adhere to Verbat’s internal document standards.
* Acceptance criteria shall be based on the clauses which were mutually discussed between Verbat and client at the Requirement Analysis phase. The same will be documented and approved by both parties through official emails.
* In the event that the Client requires any extension of the proposed acceptance schedule, the associated effort and cost of such extension will be mutually reviewed.
* If the project needs to be put on Hold / Stopped, a minimum notice period of 1 week is required along with the duration of the holding period. Thereafter Verbat will make a final decision based on the request.
* If deployment is done in the client’s server, Verbat will not be held responsible for any performance issues arising due to hardware malfunctions.
* The Client is responsible for all data-backups in case the application is not hosted on the Verbat server.
* All source codes will only be delivered or uploaded on the production server after the due payments are made to Verbat.

## GENERAL ADMINISTRATIVE, TECHNICAL & FUNCTIONAL ASSUMPTIONS

* Detailed system study is required before the start of the project.
* During the requirement gathering phase, authorized personnel from the Client’s side are expected to be available for discussions and finalization of the HLD (High Level Design), before the development commences.
* Type of reports, formats if under the scope of the project, need to be specified by the Client before the project sign off.
* Workflows if under the scope of the project, need to be specified/ confirmed by client before project signoff.
* Verbat assumes that all sign-offs from the Client will be provided within the agreed and specified timeframe.
* Client will provide all the necessary contents, both text and imagery, before starting the project in the format suggested by Verbat (if any).
* The client should provide the relevant information and data, well in advance of the execution of the related activity. Non- availability of this information or data may lead to an interruption of work, which may result in a delay in delivery as well as additional costs to the client.
* The Client should possess a server with the technical specifications recommended by Verbat for the proposed application.
* The Client will be provided with a one-time training (train the trainer) on how to use the application via a video conference (maximum of 4 hours). Additional training requests will be charged at cost to the client.
* Cost of all third-party components to be borne by the client.
* Application will support in three languages which are supported by Google Text to Speech API.

# **FINANCIALS**

## Curriculum Management Application

|  |  |  |
| --- | --- | --- |
| Sl. | Description | Amount (USD) |
| 01. |  | XXXX USD |

*Note:*

* *The above cost is exclusive of VAT applicable in UAE*
* *The above cost includes the cost of third-party service of non-translated calls.*
* *The above cost does not include third-party API purchases.*
* *The above estimate is based on the initial understanding of the requirement grounded on the details shared by client via meetings and tele-conversations. If Verbat finds during the requirement gathering phase that the actual requirements uncovered are different form the original, this might affect the estimates provided and Verbat shall exercise rights to renege on the contract. Clarifications/Changes in the scope during system study/ analysis may call in for additional effort and timeline.*
* *For feature additions, please refer section titled “Change Management “.*
* *LPO to be raised in the name of “Verbanet Technologies LLC” for project initiation*

## PAYMENT TERMS

* % of the proof of concept as advance.
* % of proof of concept on delivery of POC
* % of Language Translation Application as advance.
* % of Language Translation Application on completion of:
* SRS
* Prototype
* % on completion of development in UAT release of the Web Application on Verbat server/Mobile build.
* % of Language Translation Application to be paid on UAT Signoff.

*Note:*

* *Payment should be made within 7 days from the date of invoice.*

### MODE OF PAYMENT

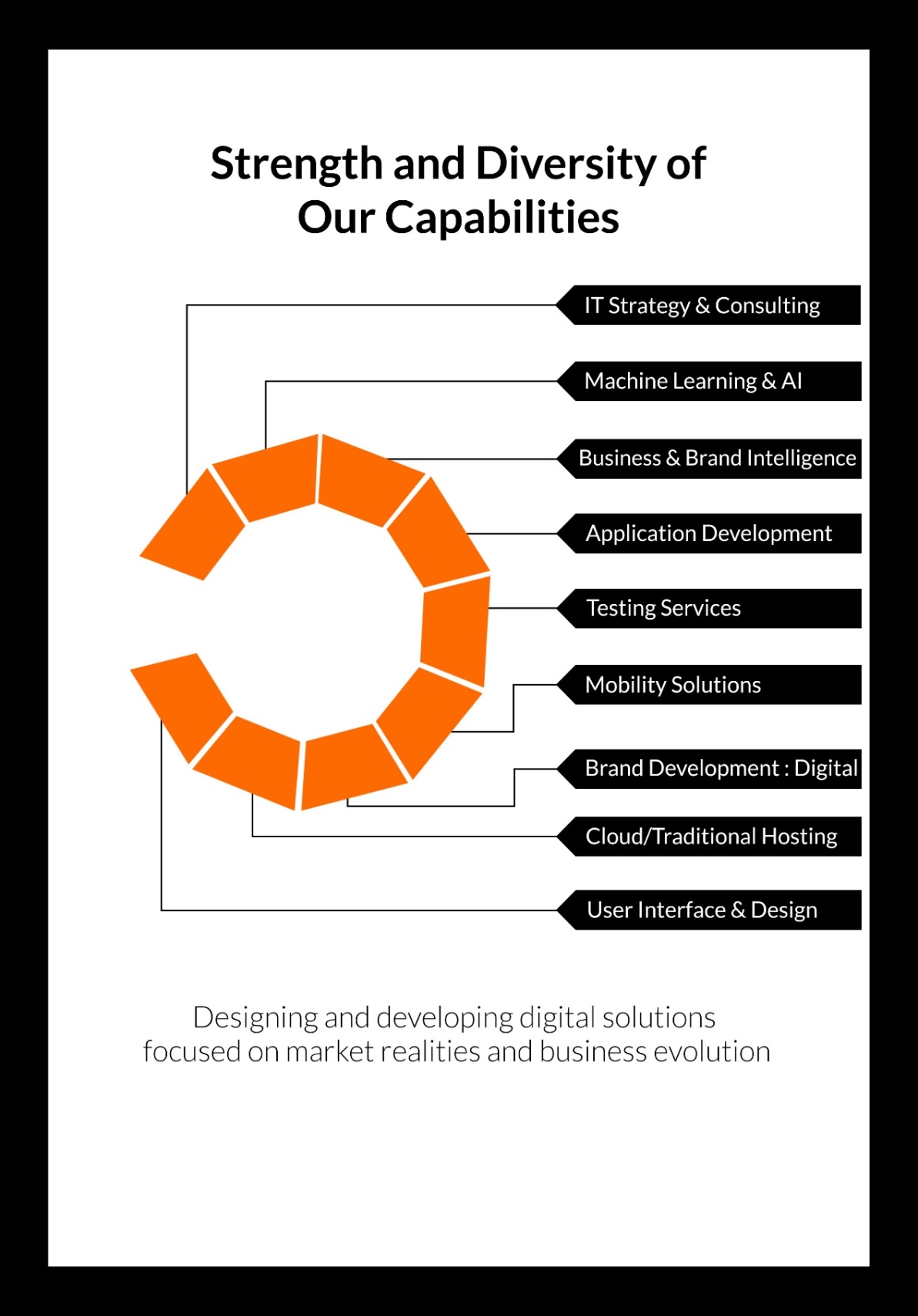
Wire transfer to our bank account

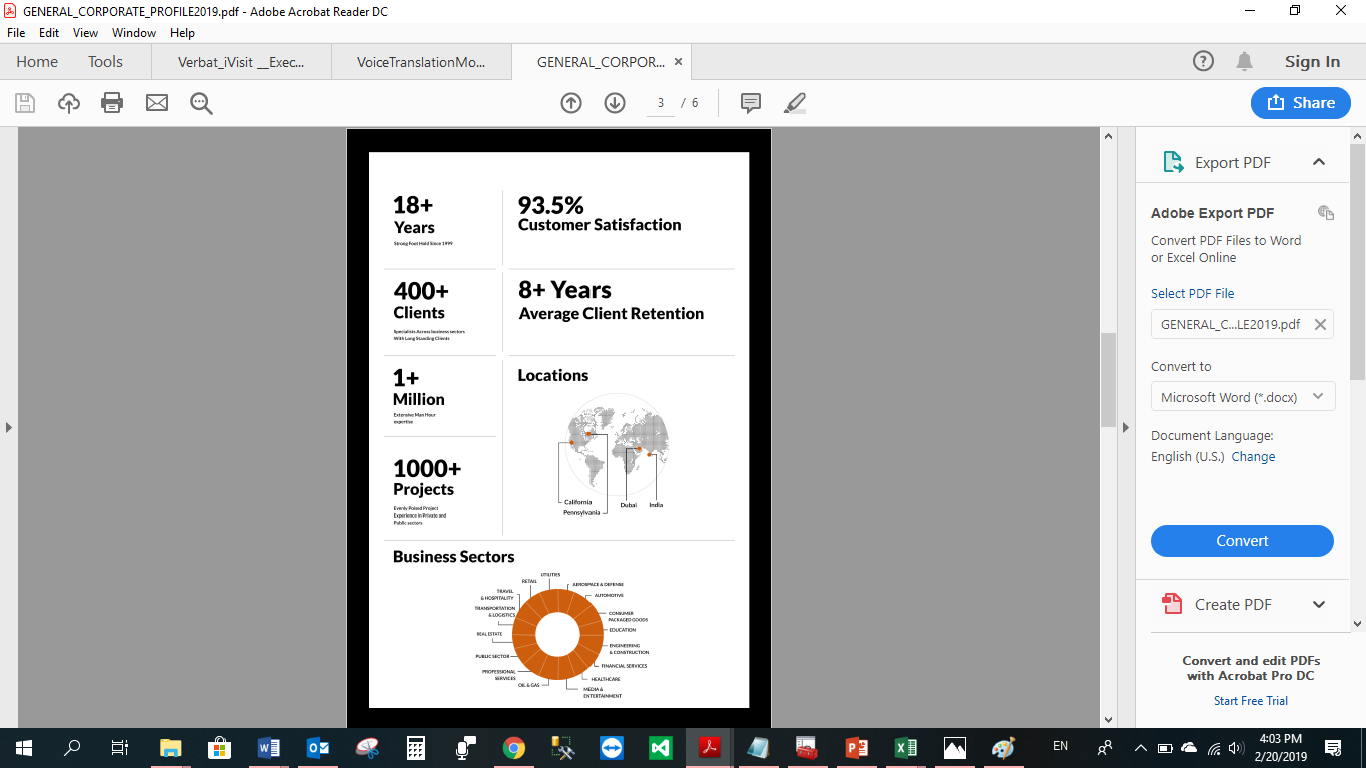
|  |  |  |
| --- | --- | --- |
| Particulars | **Option 01** | **Option 02** |
| Bank Name | Commercial Bank of Dubai | Emirates NBD |
| Account Name | Verbanet Technologies LLC | Verbanet Technologies LLC |
| Account Number | 1000305555 | 1011492858201 |
| Swift Code | CBDUAEAD | EBILAEAD |
| IBAN Number | AE840230000001000305555 | AE61 0260 0010 1149 2858 201 |
| Bank Address | Al Qusais Branch, P.O Box 2668, Al Qusais | Mamzar Branch, Dubai |

*Note:*

* *Bank charges incurred during wire transfer to be borne by the client.*
* *Any local taxes / VAT applicable to be borne by the client.*
* *Client invoices will include VAT charges in addition to the application cost.*

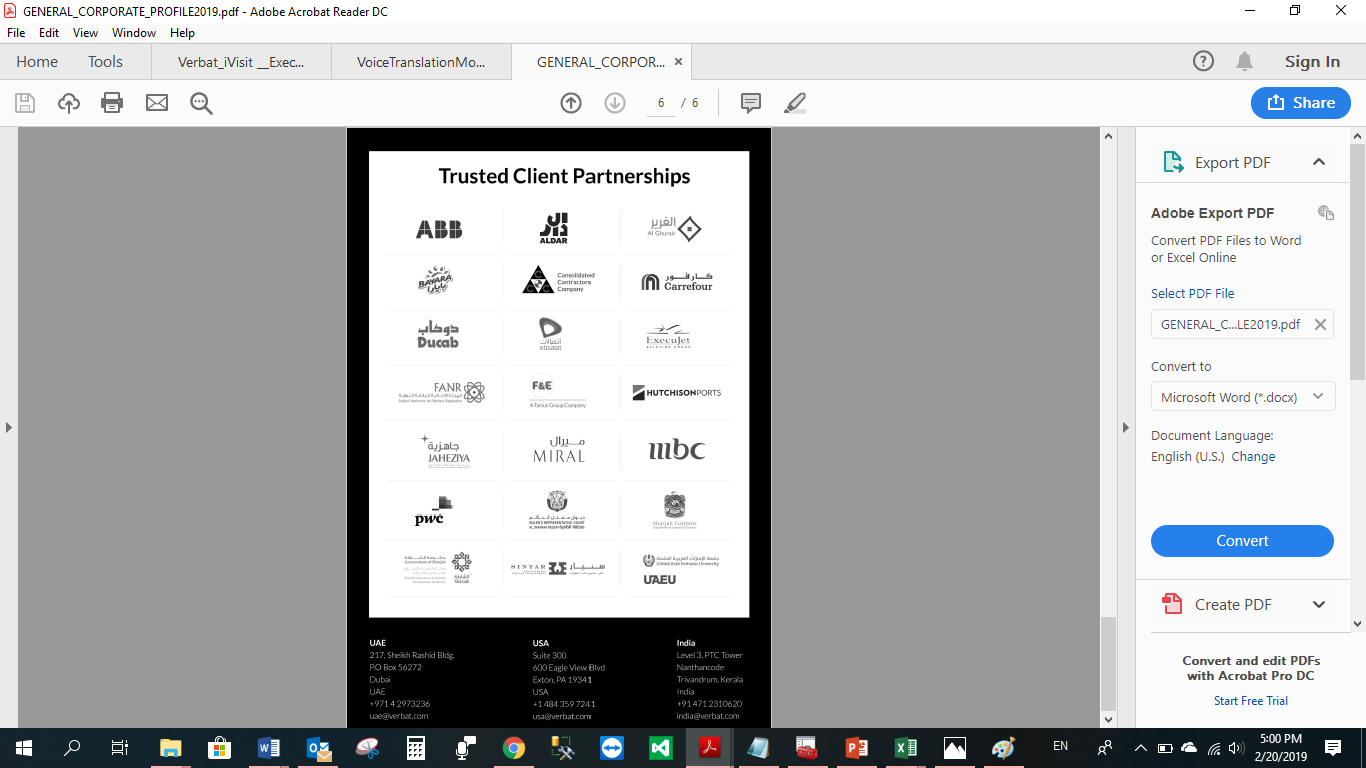












**WITH OUR**

**SINCERE THANKS.**

CONFIDENTIALITY NOTICE:

This Proposal is conﬁdential and contains proprietary information and intellectual property of Verbanet Technologies LLC. Neither this proposal nor any of the information contained herein may be reproduced or disclosed under any circumstances without the express written permission of Verbanet Technologies L.L.C (hereby known under its brand name of Verbat in the United Arab Emirates). 2019. All Rights Reserved.