

Define a MITA Design for Provider Enrollment Screening Portal System Design Specification

Author	Revision Number	Date

Define a MITA Design for Provider Enrollment Screening Portal System Design Specification	1
System Design Specification.....	3
1. Introduction	3
1.1 Purpose	3
1.2 Glossary.....	3
2. Logical Architecture	4
2.1 User Interface	5
2.2 Data Access.....	5
3. Modular Architecture.....	5
3.1 MediCare Provider Enrollment Screening Portal Frontend Module	6
3.2 MediCare Provider Enrollment Screening Portal SOAP Services Module	Error! Bookmark not defined.
3.3 MediCare Provider Enrollment Screening Portal Services Module	6
3.4 MediCare Provider Enrollment Screening Portal Miscellaneous Module	6
3.5 MediCare Provider Enrollment Screening Portal Learning Trends Module	6
3.6 Module MediCare Provider Enrollment Screening Rules Part1 Module	6
3.7 MediCare Provider Enrollment Screening Rules Part2 Module	6
3.8 MediCare Provider Enrollment Screening Rules Part3 Module	7
4. Data Model.....	7
5. Process Architecture.....	21
5.2 Persistence	21
5.3 Concurrency and Transaction Management	26
5.4 Threading.....	26
5.5 Configuration	26
5.6 Logging	27
5.7 JSP and JSF Managed Beans	28
5.8 JSF Configuration Files	28
5.9 Input Validation	28
5.10 Paginated Results	28
5.11 Risks	28
6 Security	28
6.1 Authentication	28
6.2 Authorization.....	29
6.3 Data Privacy	29
6.4 Data Integrity	29
6.5 Data Migration	29
7 Deployment and Development Concerns	29
7.1 Logical Deployment	29
7.2 Building an Enterprise Service Bus using Web Sphere ESB	30
7.3 Metrics Requirements.....	30
7.3.1 Page Performance Requirements.....	30
7.4 Technologies	30
7.4.1 Softwares.....	30
7.4.2 Database	30
7.4.3 Web Server.....	30
7.4.4 Application Server	30
7.4.5 Frameworks.....	31
7.4.6 Unit Testing	31
7.4.7 Operating System.....	31
7.4.8 Web technology.....	31
7.5 User Interface Implementation	31
7.5.1 Prototype conversion.....	31
7.5.2 Resolution.....	31

System Design Specification

1. Introduction

1.1 Purpose

The United States Center for Medicaid Services ("CMS") is running a Challenge (a series of TopCoder contests) through the NASA Center of Excellence for Collaborative Innovation (CoECI) and TopCoder to develop a web-based portal to support provider enrollment and screening. Medicaid is a healthcare program that is funded by the US federal government and by individual states. It is operated by individual states. The portal developed in this Challenge will allow medical service providers to register themselves in their individual state for compensation from the Medicaid program. While this alone will provide a very welcome new procedure for providers, who must often fill out many paper forms today, the true goal of the portal will be to automatically screen potential providers for fraud risk before they are approved for Medicaid payments. The portal will assign a potential level of risk to each registrant, and will do this by applying a suite of dynamic business rules. Once a level of risk has been assigned, the registrant's application will be sent to the appropriate sub-system for additional screening and processing.

To give a sense of the scope of the *overall project*, the Community will need to accomplish the following in this series of contests:

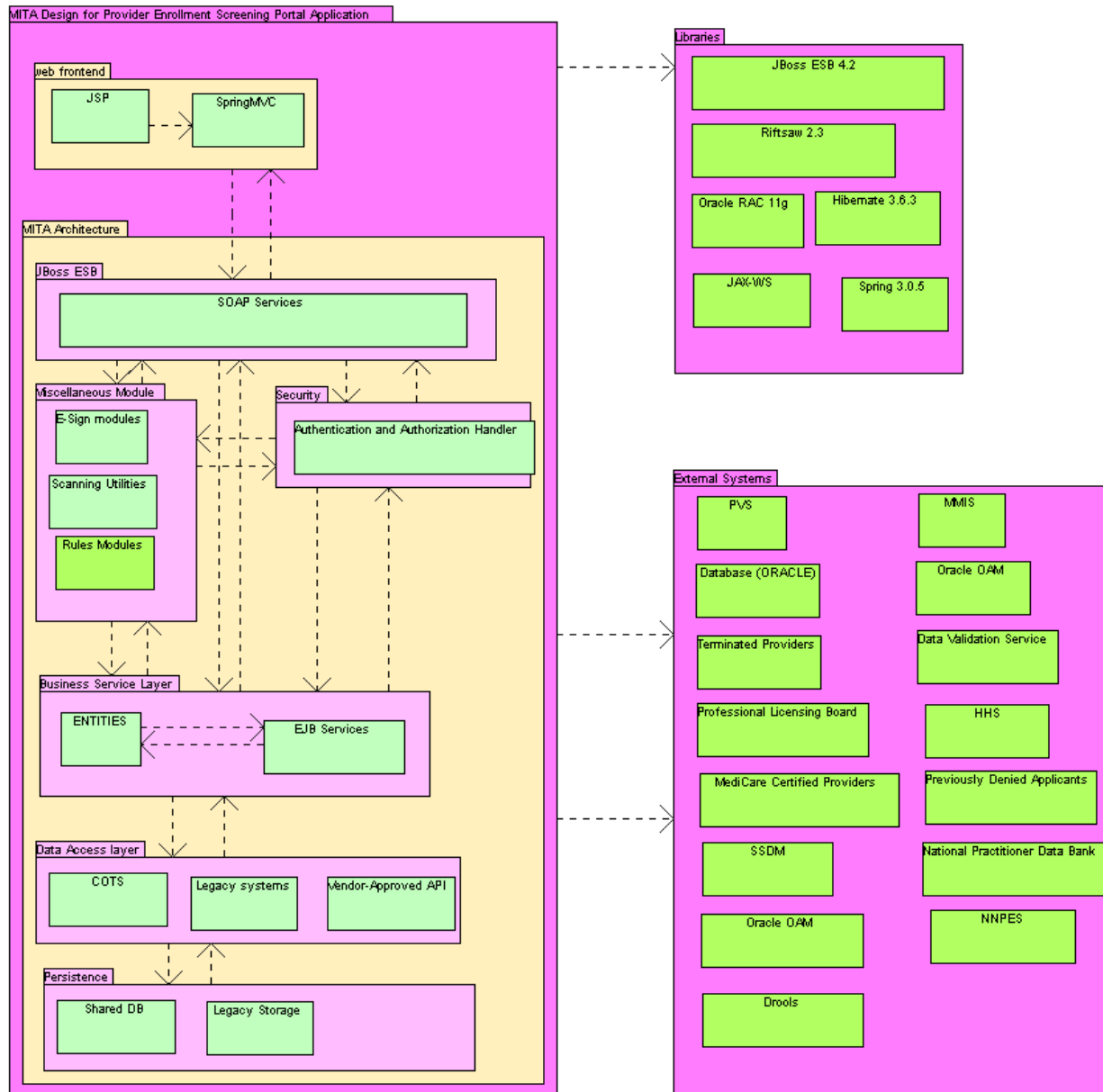
- Integrate existing business rules for provider screening, and generate new ones. This will require feedback from client subject matter experts (in a later contest, not this one).
- Design and develop the web based portal
- Design and develop the data validation and provider screening verification tiers
- Ensure the design is compliant with the MITA 3.0 Architecture (will require an ESB based architecture)
- Ensure the application complies with Section 508 requirements

1.2 Glossary

MITA	The Medicaid Information Technology Architecture. Reference: http://www.medicaid.gov/Medicaid-CHIP-Program-Information/By-Topics/Data-and-Systems/Medicaid-Information-Technology-Architecture-MITA.html
ESB	Enterprise Service Bus. Reference: http://en.wikipedia.org/wiki/Enterprise_service_bus
HIPPA	Health Insurance Portability and Accountability Act of 1996
NNPES	National Plan and Provider Enumeration System) (CMS) (only if applicable)
GUI	Graphical User Interface
CHIP	Children's Health Insurance Program Reference: http://en.wikipedia.org/wiki/State_Children's_Health_Insurance_Program
ID	Identification
MPSE	Medicaid Provider Screening and Enrollment.
LEIE	List of Excluded Individuals and Entities
NASA	National Aeronautics and Space Administration
SSDM	Social Security Death Master
CMS	Center for Medicaid Services
DROOLS	Rules Engine Technology by JBoss

2. Logical Architecture

MITA Design for Provider Enrollment Screening Portal Logical Architecture Diagram



The architecture is based on service-oriented approach with a decoupling of the web components (JSTL with Spring controllers) and the workflow components which specifically utilize an ESB (JBoss ESB) infrastructure with a workflow engine (Riftsaw). In addition, a rules engine (Drools) serves as the central repository of application rules that govern the overall processing.

The rules engine (Drools) contains the entire rules surround the questions and answers in order to determine which questions apply to any particular provider. The Drools Guvnor (<http://www.jboss.org/drools/drools-guvnor.html>) will be used as the Rules Management System.

This architecture is aligned with the CMS MITA architecture. Please refer to CMS_Challenge_PreDesign_Architecture_v20120307.doc for more details.

This application is designed to follow MITA architecture. This application will be divided in three major tiers namely presentation, business and data access.

Presentation layer consists of set of spring controllers and JSP+HTML pages which interacts with [JBoss ESB](#). JBoss ESB has support for [SOAP Web Services](#) implementations. The spring controllers take user input, constructs SOAP request message and sends it to ESB SOAP Service layer. Use input will be a form data or import files like PDFs, Scanned JPGs etc.

Business layer consists of Security, EJB and Utility implementations for interacting with data access layer and external systems. Utility classes are specifically used for implementing screening rules, capturing data from scanned copy of certificates and other. Business process is layer mainly responsible for performing the actual business logic which includes performing user authentication and authorization, CRUD operations on model entities, interacting with COTS, Legacy systems, vendor API and external systems etc..

Data Access layer responsible for performing CRUD operations against data stores. It consists of COTS, Legacy Systems and Vendor API to perform operations on share data store and legacy storage. All data store results will be modeled into respective entities and returned to calling business process layer methods.

2.1 User Interface

The frontend technology for this application was not strictly defined by the client, just a need for a simple, fast and easily maintainable solution was provided. Therefore, usage of simple JSP pages with spring mvc framework is suggested as a core technology. Spring controllers will be provided for user-driven requests' handling. Those handlers will communicate with the business level services (plain Java objects) as needed. The layer of user interface controllers will simply redirect user actions to the business services layer with. Proper mapping between controllers and JBoss ESB layer services will be designed. No business actions should be performed in user interface handlers. Dependency injection can be used to map handlers to the business level services.

The ESB layer level services will be provided during module architecture and the user interface handlers can be simply created on top of those services.

Asynchronous requests can easily handled using any javascript frameworks like jQuery. The needed controllers can be also designed to handle that asynchronous request.

2.2 Data Access

Data will be managed using data stores and legacy storage solely by the data access layer. The application uses COTS, Legacy Systems and Vendor API to manage data against data store. This layer is also responsible for caching query results, making use of DB indexes and views so as to improve the performance.

It is yet to known how COTS, Legacy Systems and Vendor API can be accessed. Time being it is assumed to use hibernate manage data in these data stores. Some sort of caching is needed for improving performance when accessing the frequently used content.

3. Modular Architecture

This application will have six modules:

- MediCare Provider Enrollment Screening Portal Frontend Module

MediCare Provider Enrollment Screening Portal Services Module
MediCare Provider Enrollment Screening Portal Miscellaneous Module
MediCare Provider Enrollment Screening Portal Learning Trends Module
MediCare Provider Enrollment Screening Rules Part1 Module
MediCare Provider Enrollment Screening Rules Part2 Module
MediCare Provider Enrollment Screening Rules Part3 Module

3.1 MediCare Provider Enrollment Screening Portal Frontend Module

This module will incorporate all the GUI screens and web-pages of the application. All the needed JSF 2.0 action handlers will be also implemented in this module

3.2 MediCare Provider Enrollment Screening Portal Services Module

This module architecture is responsible for defining the EJB and Web services

3.3 MediCare Provider Enrollment Screening Portal Miscellaneous Module

This module is responsible for implementing quartz jobs, capturing fingerprints, extracting signature and other profile information from scanned documents

3.4 MediCare Provider Enrollment Screening Portal Learning Trends Module

This module is responsible for implement the capability to learn observed negative trends factor back into screening rules so as to flag suspicious enrollments early in the screening process, ensuring the ability to detect and reduce/eliminate the incidence of false positives. The application will remember previous rejected providers and reasons for rejection corresponding form fields.

Module architect is supposed to provide an effective algorithm to implement this.

3.5 Module MediCare Provider Enrollment Screening Rules Part1 Module

This module is responsible for implementing Medicare screening rules for the following provider types

- Audiologist
- Certified Professional Midwife
- Community Health Care Worker
- Clinical Nurse Specialist
- Certified Registered Nurse Anesthetist
- Chiropractor
- Podiatrist
- Licensed Marriage and Family Therapist
- Licensed Psychologist

3.6 MediCare Provider Enrollment Screening Rules Part2 Module

This module is responsible for implementing Medicare screening rules for the following provider types

- Licensed Professional Clinical Counselor
- Physician
- Nurse Practitioner

- Occupational Therapist
- Physician Assistant
- Private Duty Nurse
- Physical Therapist
- Speech Language Pathologist
- Acupuncturist

3.7 MediCare Provider Enrollment Screening Rules Part3 Module

This module is responsible for implementing Medicare screening rules for the following provider types

- Allied Dental Professional
- Certified Mental Health Rehab Prof-CPRP
- Dentist
- Hearing Aid Dispenser
- Licensed Dietician or Licensed Nutritionist
- Licensed Independent Clinical Social Worker
- Nurse Midwife
- Optometrist

4. Data Model

The data model is provided in the TCUML. The ERD is also provided to indicate what data needs to be stored in the database.

The model has been provided in the TCUML Entity Class Diagram.

The User entity encompasses the user of this application, which as of now comprises the roles of Provider, Service Agent, Service Administrator and System Administrator

The ProviderProfile entity contains basic provider information. It holds the following data

- LAST NAME -
- FIRST NAME
- MIDDLE NAME
- NPI (Type 1 only)
- SOCIAL SECURITY NUMBER
- DATE OF BIRTH
- REQUESTED EFFECTIVE DATE
- HIGHEST DEGREE EARNED - Masters or Doctorate
- DATE DEGREE AWARDED
- INDIVIDUAL PRACTITIONER'S EMAIL ADDRESS
- Type of License/Certification
- License/Certification Number
- Original Issue Date

- Renewal End Date
- Issuing State
- Individual Practitioner's Specialty - Physician Specialties, Dental Specialties, Nurse Practitioner Specialties, Private Duty Nurse Specialties, Psychology Specialty
- Individual Practitioner's Specialty Certifications – list of certifications
 - Type of Specialty,
 - Certification Number,
 - Original Issue Date,
 - Renewal End Date,
 - Issuing State
- Private Practice
- Alternate Mailing Addresses
 - ATTN
 - ADDRESS
 - CITY
 - STATE
 - ZIP CODE
 - Remittance Advice - Address1/Address2/Address3
 - Reimbursement Check - Address1/Address2/Address3
 - Provider Correspondence - Address1/Address2/Address3
 - Authorization Request Notice and Service Agreements-Address1/Address2/Address3
 - Credentials (Enrollment Status) - Address1/Address2/Address3
- Remittance Sequence - radio button and one of the following
 - Patient Account or Own Reference Number Order
 - DHS Transaction Control Number Order
 - Recipient MHCP ID Number Order
- Group Affiliation Information -
 - GROUP NAME
 - GROUP
 - NPI/UMPI
 - PRACTICE LOCATION
 - ADDRESS
 - CITY
 - STATE
 - ZIP CODE
- Disqualification Information
 - Have you ever been convicted of a criminal offense related to involvement in any program under Medicare, Medicaid, Title XX, or Title XXI in Minnesota or any other state or jurisdiction since the inception of these programs? Yes/No,
 - Have you had civil money penalties or assessments imposed under section 1128A of the Social Security Act? Yes/No
 - Have you ever been excluded or terminated from participation in Medicare, Medicaid, Children's Health Insurance Program (CHIP), or the Title XXI services program in Minnesota or any other state since the inception of these programs? YES/NO
 -
- Contact Information
 - CREDENTIALING CONTACT
 - NAME
 - PHONE NUMBER
 - FAX NUMBER

- EMAIL ADDRESS
- Provider Statement
 - PROVIDER NAME
 - TITLE

The PrivatePractice entity contains provider practice information. It holds the following data

- Do you maintain your own private practice? – YES/No
- Primary Office Location Information
- Are you employed and/or independently contracted by a group practice? - YES/NO
- PRACTICE NAME (if applicable)
- GROUP NPI/UMPI (if applicable)
- LOCATION ADDRESS (Practice location cannot be a PO Box)
- CITY
- STATE
- ZIP CODE
- COUNTY
- OFFICE PHONE NUMBER
- OFFICE FAX NUMBER
- FEDERAL TAX ID NUMBER (if applicable and you are the owner of the practice) LEGAL NAME ACCORDING TO THE IRS (do not abbreviate)
- MINNESOTA TAX ID (if applicable and you are the owner of the practice) FISCAL YEAR END (default is 12/31)

AuditRecord contains user action data.

The Enrollment entity contains provider practice information. It holds the following data

- Enrollment Start Date
- Enrollment End Date
- Denial reason
- Appeal data

The License entity contains provider license information. It holds the following data

- Description
- License issue date
- Original issue date
- Renewal end date
- Issuing US state

The GroupAffiliation entity contains provider affiliation information. It contains the group name, NPI and address information

The Address entity contains address information. It contains location, state, city, country and zipcode information.

Role is a lookup entity is used to contain user roles. Following user roles are supported

- Provider
- Service Agent
- Service Administrator
- System Administrator

EnrollmentStatus is a lookup entity. It represents the enrollment status. Following enrollment status will be supported

- Pending

- Denied
- Approved

NotificationType is a lookup entity. It represents the type of notification to be sent to user

- Denied
- Approved

Permission is an ENUM. It contains the following enumerated values

- ALLOW
- DENY

UserPermission is used to hold user permission for particular action. It contains the following fields

- Action – denotes the name of the action like create renewal, update renewal etc..
- Permission - it denotes the corresponding permission information

FileFormat is a lookup entity. It represents the file format to which the enrollment data can be export. Presently PDF format need to be supported.

Specialty is a lookup entity. It represents the provider specialty code. Following specialty codes will be supported

Code	Description
001	GENERAL PRACTICE
002	FED QUALIFIED COMM HLTH CLIN
003	FED QUALIFIED MIGRANT HLTH CL
004	FED QUALIFIED CLINIC HOMELESS
005	EMERGENCY SERVICES
006	DIABETES
007	INPATIENT HOSPITAL PEDIATRIC
008	INPATIENT HOSPITAL PSYCH
009	INPATIENT HOSPITAL IMD
010	STATE PPO
011	ALLERGY
012	CARDIOVASCULAR DISEASE
013	DERMATOLOGY
014	GASTROENTEROLOGY
015	INTERNAL MEDICINE
016	PEDIATRICS
017	PHYSICAL MEDICINE AND REHAB
018	PREVENTIVE MEDICINE
019	PULMONARY DISEASES
020	COMP OUTPATIENT REHAB FACLTY
021	CHILD PSYCHIATRY
022	NEUROLOGY
023	PSYCHIATRY
024	PRE-PAID HEALTH PLAN, MED MAN
025	PRE-PAID HEALTH PLAN, VOLNTRY
026	PRE-PAID HEALTH PLAN, GAMC

027	REGIONAL TRT CTR-DEV DISAB
028	REGIONAL TREATMENT CTR, IMD
029	REGIONAL TRT CTR, MENTAL ILL
030	REGIONAL TRT CTR, NURSE FACIL
031	PATHOLOGY
032	RADIOLOGY
033	GERIATRIC NURSE PRACTITIONER
034	PEDIATRIC NURSE PRACTITIONER
035	FAMILY NURSE PRAC
036	LPN, LICENSED PRACTICAL
037	NURSE PRACTITIONER, ADULT
038	NURSE, REGISTERED
039	COMMUNITY HEALTH CLINIC
040	PUBLIC HEALTH CLINIC
041	ANESTHESIOLOGY
042	ENDOCRINOLOGY
043	GERIATRICS
044	IMMUNOLOGY
045	NUTRITION
046	OBSTETRICS
047	GYNECOLOGY
048	RURAL HEALTH CLINIC FREE STND
049	RURAL HEALTH CLINIC PROV BSED
050	COLON AND RECTAL SURGERY
051	GENERAL SURGERY
052	NEUROLOGICAL SURGERY
053	OBSTETRICS AND GYNECOLOGY
054	OPHTHALMOLOGY
055	ORTHOPEDIC SURGERY
056	OTOLARYNGOLOGY
057	PLASTIC SURGERY
058	THORACIC SURGERY
059	UROLOGY
060	ONCOLOGY
061	ORAL SURGEON (DENTISTS)
062	DENTAL-GENERAL PRACTITIONER
063	ORTHODONTIA
064	PEDODONTIST
065	PERIODONTICS
066	INDEPENDENT DIAGNOSTIC TESTING
067	PHARMACY, HOME CARE
068	PHARMACY, HOSPITAL
069	PHARMACY, NUCLEAR
070	PHARMACY, COMMUNITY
071	PHARMACY, NURSING FACILTY
072	INFECTIOUS DISEASES

073	PROSTHODONTICS
074	PERIPHERAL VASCULAR DIS/SURG
075	PREVENTIVE MEDICINE
076	RADIOLOGY & RADIATION THERAPY
077	FAMILY PRACTICE
078	FAMILY COUNSELING
079	FAMILY TRAINING
080	NURSE PRACTITIONER OBGYN
081	NEUROPSYCHOLOGY
082	NURSE MENTAL HEALTH
083	PERSONAL CARE ATTENDANT
084	SERV FOR CHILD W/HANDICAP PRV
085	SER FOR CHLD W/HANDICAP FAM M
086	MEDICAL TRANSPORTATION - AIR
087	MEDICAL TRANSPORTATION - ALS
088	MEDICAL TRANSPORTATION - BLS
089	MEDICAL TRANSPORTATION - SPEC
090	RHEUMATOLOGY
091	CARDIOVASCULAR SURGERY
092	NEPHROLOGY
093	JOINT COMMISSION ACCREDITED
094	NURSE PRACTITIONER NEONATAL
095	PCA CHOICE
096	NURSE PRACTITIONER WOMENS HLTH
097	NURSE PRACTITIONER ACUTE CARE
098	PSYCH 21 OPTION
099	OTHER
AB	ADULT DAY CARE, BATH
AD	ADULT DAY CARE SERVICES
AL	CD ADOLESCENT
AP	ACUPUNCTURE CERTIFIED
AS	TRIBAL ASSURANCE STATEMENT
AT	ASSISTANT TECHNOLOGY
AV	CAREGIVER ASSESSMENT
BA	BILLING AGENT
BC	CDCS BACKGROUND CHECK
BD	BEHAVIOR PROGRAM BY AIDE
BN	BEHAVIOR PROGRAM BY ANALYST
BP	BEHAVIOR PROGRAM BY PROFESSION
BS	BEHAVIOR PROGRAM BY SPECIALIST
BT	BEHAVIORAL THERAPY
C2	CD CO-OCCURRING
CA	CASE MANAGEMENT AIDE
CB	CONSOLIDATED BILLING AGENCY
CC	CRITICAL CARE
CD	CONSUMER DIRECTED COMM SUPP

CE	CONSUMER TRAINING AND EDUCATI
CF	CUSTOMIZED LIVING CLASS F
CG	24HR CUSTOMIZED LIVING CLASS F
CH	24 HOUR CUSTOMIZED LIVING CL A
CI	CONTRACTED INDIVIDUAL
CL	CUSTOMIZED LIVING CLASS A
CM	CDCS MANDATORY CASE MGT
CN	COMPANION CARE/SERVICES
CO	CASE MANAGEMENT ONGOING
CP	CRISIS RESPITE, SPECIALIZED
CR	CRISIS RESPITE
CS	CHORE SERVICES
CT	CAREGIVER TRAINING AND EDUCAT
CV	CASE MANAGEMENT CONVERSION
CW	CD CLIENT WITH CHILD
DA	DT&H PILOT RATE A
DB	DT&H PILOT RATE B
DC	DT&H PILOT RATE C
DD	DT&H PILOT RATE D
DO	AC DISCRETIONARY SERVICES
DS	MED SUPPLY DIALYSIS
DT	DT&H TRANSPORTATION
EA	24HR EMERGENCY ASSISTANCE
ED	ENDODONTISTS
EV	ELIGIBILITY VERIFICATION VEND
FA	FOSTER CARE CORPORATE
FC	FAMILY COUNSELING
FF	FOSTER CARE ADULT FAMILY
FK	FOSTER CARE CHILD
FL	FOND DU LAC INDIAN RESRV
FT	FAMILY TRAINING
GP	GRAND PORTAGE INDIAN RESRV
GT	GAMBLING TREATMENT
HA	HOUSING ACCESS COORDINATION
HB	CD HOSPITAL BASED INPATIENT
HD	HOME DELIVERY MEALS
HF	24HR CUSTOMIZED CLASS F FADS
HH	HOME HEALTH - AIDE AND EXTEND
HI	HOME INFUSION PHARMACY
HM	HOMEMAKER SERVICES
HS	24HR CUSTOMIZED CLASS A FADS R
IC	INDEP LIV SKILLS, COUNSELING
IF	IN-HOME FAMILY SUPPORT
IG	INDEP LIV SKILLS, GRP THERAPY
II	INDEP LIV SKILLS, IND THERAPY
IM	INDEP LIV SKILLS, MAINTENANCE

IN	PERSONAL INSTALLATION
LC	LIVE-IN CAREGIVE EXPENSES
LL	LEECH LAKE INDIAN RESRV
LS	LOWER SIOUX
M1	CD MEDICATION ASST THER-METH
M2	CD MEDICATION ASST THER-OTHER
M3	CD MEDICATN ASST THER-METH+
M4	CD MEDICATN ASST THER-OTHER+
MC	MEDICARE CARRIER
MD	MODIFICATIONS/ADAPTATIONS
ML	MILLE LACS INDIAN RESRV
MS	CD MEDICAL SERVICE
ND	DONT-PUBLISH-IN-DIRECTORY
NL	NETT LAKE INDIAN RESRV
NP	DT&H PILOT WAIVER NON-PILOT
NR	CD NON-RESIDENTIAL TREATMENT
NS	NIGHT SUPERVISION
NT	NUTRITIONAL SERVICES
OA	OCCUPATIONAL THERAPY,ASST EXT
OT	OCCUPATIONAL THERAPY, EXTENDED
PA	PHYSICAL THERAPY,ASST EXTENDED
PC	PERSONAL CARE PROVIDER ORG
PD	RN AND LPN, NON-EXTENDED
PI	PRAIRIE ISLAND INDIAN RESRV
PR	PRIMARY CARE
PS	PERSONAL SUPPORT
PT	PHYSICAL THERAPY, EXTENDED
PU	PERSONAL PURCHASE
PV	PREVOCATIONAL SERVICES
QS	QUALIFIED SUPERVISOR
R1	CD RESIDENTIAL TREATMENT-HIGH
R2	CD RESIDENTIAL TREATMENT-MED
R3	CD RESIDENTIAL TREATMENT-LOW
RB	CD RM & BOARD W/TRMT ONE LOCAT
RC	RESPIRE, CERTIFIED FACILITY
RH	RESPIRE, HOSPITAL
RI	RESPIRE CARE SERVICES, IN HOME
RL	RED LAKE INDIAN RESRV
RM	CD RM & BOARD W/TRMT TWO LOCAT
RO	RESPIRE CARE SERVICES,OUT HOME
RS	RESIDENTIAL CARE SERVICES
RT	RESPIRATORY THERAPY, EXTENDED
S1	CD SPECIAL POPULATIONS
SA	SLS, CHILD
SC	SLS, ADULT, CORPORATE AFC
SD	STRUCTURED DAY

SE	SUPPORTED EMPLOYMENT
SL	SUPPORTED LIVING SVCS, ADULT
SM	SHAKOPEE MDEWAKANTON SIOUX
SN	SKILLED NURSING
SP	NOT APPLICABLE
SR	PERSONAL SERVICE FEE
SS	SPECIALIST SERVICE
ST	SPEECH THERAPY, EXTENDED
SU	SUPPLIES/ EQUIPMENT (AC)
SX	SUPPLIES/ EQUIPMENT EXTENDED
TA	TRANSPORTATION - EXTRA ATTD
TD	TELEMEDICINE
TE	TELEHOMECARE
TF	TRANSITIONAL SVCS,FURNITURE
TH	TRANSITIONAL SVCS,HOUSELD FURN
TL	TELERADIOLOGY
TM	TRANSPORTATION - MILEAGE COM
TN	TRANSPORTATION - MILEAGE (NON)
TP	HEALTHCARE DATA TRADING PART
TR	TRANSPORTATION - AC 1 WAY TRIP
TS	TRANSITIONAL
TW	TRANSPORTATION - WAIVER TRIP
US	UPPER SIOUX INDIAN RESRV
WC	WHEELCHAIR TRANSPORT
WE	WHITE EARTH INDIAN RESRV

HelpItem this entity contains the help information about the application. It contains help title and description

UserSearchCriteria contains user search criteria data used to search users. It contains the following criteria items.

- User name
- Provider type
- Page number
- Page size
- Status – denotes enrollment status.
- andOrFlag – if false, returns search results satisfying any of the criteria term.
- Full – if true, return search results with aggregate data loaded fully

HelpSearchCriteria contains search criteria data used to search help items. It contains the following.

- term – search term

SearchResult<T> is a utility entity used to return paged search results. It provides the requested page number and size. But also provides the total amount of items that qualify for the request (which may have been filtered) and also provides the total page count, which is a calculated value from the previous two values of the (total page/page size) rounded up. And of course, it provides the list of the requested items that satisfy the requested page.

Audit and logging functionality will require to format and output strings with input/output information about data entities. All data entities will provide a JSON version of their data, to be supplied in a `toJsonString():String` method. This will standardize the logging of input and out information of entities. Jackson Java JSON-processor will be used to generate JSON version of entities.

Overall, these entities are geared to be used as data transport objects. They will be simple JavaBeans with getters and setters, and no intrinsic business validation in themselves.

5. Conceptual Data Model (CDM)

The conceptual data model of the data model in TCUML Entity Class Diagram is shown in the following table

Model Class Name	Business Process	Super Class
User	The User entity encompasses the user of this application, which as of now comprises the roles of Provider, Service Agent, Service Administrator and System Administrator	IdentifiableEntity
ProviderProfile	<p>The entity contains basic provider information. It holds the following data</p> <ul style="list-style-type: none"> ○ LAST NAME - ○ FIRST NAME ○ MIDDLE NAME ○ NPI (Type 1 only) ○ SOCIAL SECURITY NUMBER ○ DATE OF BIRTH ○ REQUESTED EFFECTIVE DATE ○ HIGHEST DEGREE EARNED - <ul style="list-style-type: none"> ○ Masters or Doctorate ○ DATE DEGREE AWARDED ○ INDIVIDUAL PRACTITIONER'S EMAIL ADDRESS ○ Type of License/Certification ○ License/Certification Number ○ Original Issue Date ○ Renewal End Date ○ Issuing State ○ Individual Practitioner's Specialty - <ul style="list-style-type: none"> ○ Physician Specialties, Dental Specialties, Nurse Practitioner Specialties, Private Duty Nurse Specialties, Psychology Specialty ○ Individual Practitioner's Specialty Certifications – list of certifications <ul style="list-style-type: none"> ▪ Type of Specialty, ▪ Certification Number, ▪ Original Issue Date, ▪ Renewal End Date, ▪ Issuing State ○ Private Practice ○ Alternate Mailing Addresses <ul style="list-style-type: none"> ▪ ATTN ▪ ADDRESS ▪ CITY ▪ STATE ▪ ZIP CODE 	IdentifiableEntity

		<p>Remittance Advice - Address1/Address2/Address3 Reimbursement Check - Address1/Address2/Address3 Provider Correspondence - Address1/Address2/Address3 Authorization Request Notice and Service Agreements- Address1/Address2/Address3 Credentials (Enrollment Status) - Address1/Address2/Address3</p> <ul style="list-style-type: none"> ○ Remittance Sequence - radio button and one of the following <ul style="list-style-type: none"> ▪ Patient Account or Own Reference Number Order ▪ DHS Transaction Control Number Order ▪ Recipient MHCP ID Number Order ○ Group Affiliation Information - <ul style="list-style-type: none"> ▪ GROUP NAME ▪ GROUP ▪ NPI/UMPI ▪ PRACTICE LOCATION ▪ ADDRESS ▪ CITY ▪ STATE ▪ ZIP CODE ○ Disqualification Information <ul style="list-style-type: none"> ▪ Have you ever been convicted of a criminal offense related to involvement in any program under Medicare, Medicaid, Title XX, or Title XXI in Minnesota or any other state or jurisdiction since the inception of these programs? Yes/No, ▪ Have you had civil money penalties or assessments imposed under section 1128A of the Social Security Act? Yes/No ▪ Have you ever been excluded or terminated from participation in Medicare, Medicaid, Children's Health Insurance Program (CHIP), or the Title XXI services program in Minnesota or any other state since the inception of these programs? 	
--	--	--	--

	<p>YES/NO</p> <ul style="list-style-type: none"> ▪ ○ Contact Information <ul style="list-style-type: none"> ▪ CREDENTIALING CONTACT NAME ▪ PHONE NUMBER ▪ FAX NUMBER ▪ EMAIL ADDRESS ○ Provider Statement <ul style="list-style-type: none"> ▪ PROVIDER NAME ▪ TITLE ○ 	
PrivatePractice	<p>The entity contains provider practice information. It holds the following data</p> <ul style="list-style-type: none"> ▪ Do you maintain your own private practice? – YES/No ▪ Primary Office Location Information ▪ Are you employed and/or independently contracted by a group practice? - YES/NO ▪ PRACTICE NAME (if applicable) ▪ GROUP NPI/UMPI (if applicable) ▪ LOCATION ADDRESS (Practice location cannot be a PO Box) ▪ CITY ▪ STATE ▪ ZIP CODE ▪ COUNTY ▪ OFFICE PHONE NUMBER ▪ OFFICE FAX NUMBER ▪ FEDERAL TAX ID NUMBER (if applicable and you are the owner of the practice) LEGAL NAME ACCORDING TO THE IRS (do not abbreviate) ▪ MINNESOTA TAX ID (if applicable and you are the owner of the practice) FISCAL YEAR END (default is 12/31) 	IdentifiableEntity
AuditRecord	This entity contains user action data.	IdentifiableEntity
Enrollment	<p>The Enrollment entity contains provider practice information. It holds the following data</p> <ul style="list-style-type: none"> • Enrollment Start Date • Enrollment End Date • Denial reason 	IdentifiableEntity

	<ul style="list-style-type: none"> • Appeal data 	
License	<p>The License entity contains provider license information. It holds the following data</p> <ul style="list-style-type: none"> • Description • License issue date • Original issue date • Renewal end date • Issuing US state 	LookupEntity
GroupAffiliation	<p>The GroupAffiliation entity contains provider affiliation information. It contains the group name, NPI and address information</p>	LookupEntity
Address	<p>The Address entity contains address information. It contains location, state, city, country and zipcode information.</p>	IdentifiableEntity
	<p>The Address entity contains address information. It contains location, state, city, country and zipcode information.</p>	IdentifiableEntity
Role	<p>Role is a lookup entity is used to contain user roles. Following user roles are supported</p> <ul style="list-style-type: none"> • Provider • Service Agent • Service Administrator • System Administrator 	LookupEntity
EnrollmentStatus	<p>EnrollmentStatus is a lookup entity. It represents the enrollment status. Following enrollment status will be supported</p> <ul style="list-style-type: none"> • Pending • Denied • Approved 	LookupEntity
NotificationType	<p>NotificationType is a lookup entity. It represents the type of notification to be sent to user</p> <ul style="list-style-type: none"> • Denied • Approved 	LookupEntity
Permission	<p>Permission is an ENUM. It contains the following enumerated values</p> <ul style="list-style-type: none"> • ALLOW • DENY 	NA
UserPermission	<p>UserPermission is used to hold user permission for particular action. It contains the following fields</p> <ul style="list-style-type: none"> • Action – denotes the name of the action like create renewal, update renewal etc.. • Permission - it denotes the corresponding permission information 	IdentifiableEntity
FileFormat	<p>FileFormat is a lookup entity. It represents the file format to which the enrollment data can be export. Presently PDF</p>	LookupEntity

	format need to be supported.	
Specialty	Specialty is a lookup entity. It represents the provider specialty code. Following specialty codes will be supported	IdentifiableEntity
HelpItem	HelpItem this entity contains the help information about the application. It contains help title and description	IdentifiableEntity
UserSearchCriteria	<p>UserSearchCriteria contains user search criteria data used to search users. It contains the following criteria items.</p> <ul style="list-style-type: none"> • User name • Provider type • Page number • Page size • Status – denotes enrollment status. • andOrFlag – if false, returns search results satisfying any of the criteria term. • Full – if true, return search results with aggregate data loaded fully 	IdentifiableEntity
HelpSearchCriteria	<p>HelpSearchCriteria contains search criteria data used to search help items. It contains the following.</p> <ul style="list-style-type: none"> • term – search term 	IdentifiableEntity
SearchResult	SearchResult<T> is a utility entity used to return paged search results. It provides the requested page number and size. But also provides the total amount of items that qualify for the request (which may have been filtered) and also provides the total page count, which is a calculated value from the previous two values of the (total page/page size) rounded up. And of course, it provides the list of the requested items that satisfy the requested page.	IdentifiableEntity

6. Services

UserService: Provides the methods to manage a user. This includes creating users, updating their roles and passwords, enabling or disabling them. There are many methods to retrieve users.

ProviderProfileService: Provides the methods to manage provider profile. This includes creating, updating, deleting and retrieving provider profiles.

SecurityService This service provides a method to perform authentication and authorization.

ExportService: This service provides methods to export enrollment data into PDF format.

NPIRegistryService: This service provides methods to retrieve NPI data of providers given provider search term.

ScreeningService: This service provides methods to manage screening and scheduling screening

HelpService: A simple service used to get help files and the company's about data. It also provides ability to save new help items.

EmailNotificationService: Provides a single method to sends notification emails to users.

LookupService: A simple service that gets lookup values for various lookup entities. Examples major program, category service, tribal code, specialty code etc.,

AuditService – This service provides methods which are used for auditing purpose. The SOAP service methods will use the methods of this service to perform auditing.

UserPermissionService: Provides the methods to manage a user permissions.

7. Process Architecture

User Functionality

Login

- The user will be authenticated with the SecurityService.authenticate. This will use Oracle OAM to authenticate the user. If authentication fails, this would throw an exception.
- If authentication is successful, then it returns session token which will be used for later communication.
- The session timer is configurable at the level of the application server

Logout

- The logout process does not require any cleanup, so it would simply invalidate the user session.
- The expiration of the session timer would result in the application server also invalidating the session.

View Homepage

- This navigation page would need to provide different options based on the role found in the session. The JSF + JSP templates would be coded to check the user role and render the HTML based on this.

View Help

- The help page provides a means of searching for documents based on free text search.
- The search involves providing that search term, a page (1-based), and a page size. The page size dropdown would be configured at the controller layer level.
- The HelpService.search is used. The search will attempt to find any match of the term in the text.
- Matches are returned in a SearchResult object that contains the current page and size, the total amount of records, and the total amount of pages based on the input parameters, and this is used to render the paging navigation. And of

course it returns the matching records for the requested page. The results are displayed.

Create Enrollment

- The ProviderProfileService.create is used to create enrollment for user. It will set the enrollment.status to "new"

Edit Enrollment

- The ProviderProfileService.update is used to update enrollment for user..

View Enrollment

- The ProviderProfileService.get is used to retrieve user entity with provider profile data.

Query Enrolment Status

- The ProviderProfileService.get is used to get enrollment status. The ProviderProfile contains enrollment from which status can be retrieved.

Export Enrollment

- The ExportService.export is used to export enrollment (provider profile) data into PDF form given user identifier.

Create Renewal

- The ProviderProfileService.update is used to create renewal of enrollment.
- It will set the enrollment.status to "renewal"

Edit Renewal

- The ProviderProfileService.update is used to create renewal of enrollment.
- Note it should not update the enrollment.status

View Renewal Enrollment

- The ProviderProfileService.get is used to retrieve provider profile data.

Export Renewal Enrolment Data

- The ExportService.export is used to export renewal enrollment (provider profile) data into PDF form given user identifier.

View User Profile

- The UserService.get is used to retrieve user entity without provider profile data.

Update User Profile

- The UserService.update is used to update user data. Note it will only update user entity i.e. it can update firstname, lastname, middle name and email.

Print Form

- The UserService.get is used to retrieve user entity with provider profile data.

Service Administrator Functionality

Search Enrolments

- The ProviderProfileService.search is used to retrieve user entity with provider profile data.
- Matches are returned in a SearchResult object that contains the current page and size, the total amount of records, and the total amount of pages based on the input parameters, and this is used to render the paging navigation. And of course it returns the matching records for the requested page. The results are displayed.

Set Enrolment Status

- The UserService.setEnrollmentStatus is used to set enrollment status

Override Previously Denied Status

- The UserService.setEnrollmentStatus is used to set enrollment status
- The UserService.appeal is used to set user appeal in enrollment entity

Initiate On-Demand Screening

- The ScreeningService.performScreening is used to perform on demand screening of enrollment. i.e when the user submits enrollment for review
- The ScreeningService.scheduleMediCareProgramDataChange is used to check MediCare program data changes. i.e when screening rules changes. Handling screening rules will be done by different contests.
-

Initiate Scheduled Screening

- The ScreeningService.performScreening is used to perform on demand screening of enrollments

Initiate Re-Validation

- The ScreeningService.performScreeningByld is used to perform on revalidation given enrollment id.

View Users

- The UserService.search used to retrieve users. It will return first page with default page size set to 10.

Create User

- The UserService.create used to create username and password at ORACLE OAM
- It sets the user status to active on successful creation.

Update User

- The UserService.update used to update user password at ORACLE OAM and user email, first name, last name and middle name at local application.

Delete User

- The UserService.user used to delete user account from the application.
- It won't actually delete the user but It sets the user status to delete.

View Roles

- The RoleService.getAll used to retrieve all roles.

Create Role

- The RoleService.create used to create new role in to the application

Update Role

- The RoleService.update used to update role information.

Delete User

- The RoleService.delete used to delete role information.

Manage User Permissions

- No Details are provided by client now. It is assumed it will be available during module architecture.

Service Administrator Functionality

Send Email Notifications

- The notification will take place with the NotificationService.
- The NotificationService.send is used to send emails to users.

Perform Logging

- TC logging wrapper is used to perform logging

Perform Auditing

- The AuditService.audit is used to perform auditing. It will store audit data into database.

Integrate With External Systems

- The POCs are proposed to check the new application will integrate with existing system nicely

Perform Automatic Screening

- The actual screening is not in scope of the architecture. However it defines ScreeningService.scheduleMediCareProgramDataChange is used to changes to various external systems data. Concrete details about external systems will be available during module architecture contests.

Learn Trends for Screening

- Module architecture is proposed to handle this task. At this point the screening rules for providers are not available. After completing screening rules or data about screening rules is available then this module architecture can be run.

Generate Reports

- The ExportService.export can be used to export the application (enrollment) data into PDF format.
- The TC PDF Builder component can be used for this purpose

Schedule Re-Validation

- The ScreeningService.scheduleRevalidation can be used to schedule revalidation.

Schedule Screening

- The ScreeningService.scheduleScreening can be used to schedule screening.

Integrate With MN-ITS Mail Box

- Details about MN-ITS Mail Box is yet to be provided. It is expected that the downstream contests will handle this.

Change Application Language

- In Spring MVC application, language can be changed easily by using org.springframework.web.servlet.i18n.SessionLocaleResolver

Pass DHS Audit

- This process is done manually

Perform Background Check

- This will be handled by screening module. ScreeningRules for enrollments are yet to be finalized. A concept contests are running for this.
- The UserService.performBackgroundCheck is used to perform this action

Verify Business Ownership

- The UserService.verifyBusinessOwnership is used to perform this action. The actual business logic will be handled in screening rules modules

5.2 Persistence

Hibernate (JPA version) will be used for managing data against data stores. Making use of annotation based dependency injection in EJBs, EntityManager will be injected in the Webservices and EJBs.

So hibernate persistence should be hidden behind the services. The following approaches (both) should be implemented for Hibernate persistence. An EJB Web service session will be consumed by external systems (To be used by other US States along with Minnesota)

A local EJB session facade is created to wrap Hibernate persistence. The session facade provides adapter logic to map Hibernate entity POJOs to Service Data Objects and back. An integration developer can then use an EJB import to invoke the session facade, and invoke it in a tightly coupled fashion with corresponding Qualities of Service (QoS).

An EJB Web service session facade is created to wrap Hibernate persistence. An integration developer can then use a Web service import to invoke the Web service for persistence. This gets around having to build POJO to SDO converters, since at the current time SCA only uses SDO for data types.

5.3 Concurrency and Transaction Management

Any database operations must be transactional safe. Transaction will be managed using EJB declarative transactional control by the container.

5.4 Threading

The application will be effectively thread-safe. The entities being persisted won't change during a persistence operation.

5.5 Configuration

The front-end layer of the application can use injection for configuration with the use of Spring. For backend layers like we EJB, JBossWS spring should not be used for configurations, since it is not required.

The configuration of the SMTP server and the database will need to be referenced to JBoss JNDI settings. Here are examples of this:

Database

```
<bean id="#datasourceName#" class="org.springframework.jndi.JndiObjectFactoryBean">
  <property name="jndiName"><value>jdbc/#JBoss_DataSource_Alias#</value></property>
```

```

</bean>
Mail
<bean id="mailSession" class="org.springframework.jndi.JndiObjectFactoryBean">
    <property name="jndiName"><value>java:comp/env/#MailSessionName#</value></property>
</bean>

```

value of instances of JndiObjectFactoryBean should come from JBoss settings.

5.6 Email Notifications

JavaMail will be used to send email notification. Spring Velocity is used to get separate email templates for each notification type.

The following shows configuration for setting up velocity templates

```

<bean id="velocityEngine" class="org.springframework.ui.velocity.VelocityEngineFactoryBean">
    <property name="resourceLoaderPath"><value>/WEB-INF/templates</value></property>
</bean>

<bean id="mailSender" class="org.springframework.mail.javamail.JavaMailSenderImpl">
    <property name="host"><value>mail.xxx.org</value></property>
    <property name="username"><value>xxx</value></property>
    <property name="password"><value>xxx</value></property>
</bean>

<bean id="mailMessage" class="org.springframework.mail.SimpleMailMessage"/>

<bean id="managerTarget" class="com.xxx.MailSender">
    <property name="mailSender"><ref bean="mailSender"/></property>
    <property name="message"><ref bean="mailMessage"/></property>
    <property name="velocityEngine"><ref bean="velocityEngine"/></property>
</bean>

```

5.7 Logging

The application will log activity and exceptions using the LoggingWrapper.

It will log errors at Error level, and method entry/exit information at DEBUG level.

Specifically, logging will be performed as follows, if logging is turned on.

- Method entrance and exit will be logged with DEBUG level.
 - o Entrance format: [Entering method {className.methodName}]
 - o Exit format: [Exiting method {className.methodName}]. Only do this if there are no exceptions.
- Method request and response parameters will be logged with INFO level
 - o Format for request parameters: [Input parameters[{request_parameter_name_1}:{request_parameter_value_1}, {request_parameter_name_2}:{request_parameter_value_2}, etc.}]
 - o Format for the response: [Output parameter {response_value}]. Only do this if there are no exceptions and the return value is not void.
 - o If a request or response parameter is complex, use its toString() method. If that is not implemented, then print its value using the same kind of name:value notation as above. If a child parameter is also complex, repeat this process recursively.
- All exceptions will be logged at ERROR level, and automatically log inner exceptions as well.
 - o Format: Simply log the text of exception: [Error in method {className.methodName}: Details {error details}]

- The stack trace of the error and a meaningful message.

In general, the order of the logging in a method should be as follows:

1. Method entry
2. Log method entry
3. Log method input parameters
4. If error occurs, log it and skip to step 7
5. Log method exit
6. If not void, log method output value
7. Method exit

All entities will provide a JSON version of their data, to be supplied in a `toJSONString():String` method. This will standardize the logging of input and out information of entities.

Note the application should not log sensitive data like user password, medical data etc...

5.8 JSP and Spring Controllers

All frontend pages will be based on JSPs and, as required by the system architecture, will incorporate functionality provided by the JSTL standard tags.

5.9 Spring Configuration Files

The assembler will provide the configuration file (mainly for controllers) for web application deployment.

5.10 Input Validation

The input validation not involving business logic should be configured in the validation XML; while the validation involving business logic should be done using spring validation framework. SOAP services should also have validation.

5.11 Paginated Results

[jQuery datatable](#) can be used for showing paginated results.

5.12 Risks

Not All technologies are known. It is bit stiff task on how well this application will be integrated with diverse external systems. A proof of concept should be run to assure the application will integrates nicely with existing external systems.

6 Security

6.1 Authentication

The application will be integrated with authentication system which is built against the external directory (Oracle OAM). Oracle OAM return access token on successful authentication, this means that this application will rely on a session token that identifies the user. The web client, after successful authentication, will use the session token to perform additional operations.

```
public static final String ms_resource = "//Example.com:80/login.jsp";

public static final String ms_protocol = "http";
public static final String ms_method = "GET";
public static final String ms_login = "jsmith";
public static final String ms_passwd = "j5m1th";
public String m_configLocation = "/oracle";
```

```
AccessClient ac = AccessClient.createDefaultInstance(m_configLocation,
    AccessClient.CompatibilityMode.OAM_11G);
```

```
ResourceRequest rrq = new ResourceRequest(ms_protocol, ms_resource,
                                          ms_method);

Hashtable creds = new Hashtable();
creds.put("userid", ms_login);
creds.put("password", ms_passwd);
UserSession session = new UserSession(rrq, creds);
if(session.getStatus() == UserSession.LOGGEDIN) {

    return session.getSessionToken\(\)

}
```

6.2 Authorization

Authorization will be using session token obtained at [Authentication](#) . The following shows how to authorize user for accessing a resource URL.

```
public static final String ms_resource = "//Example.com:80/users/
                                          create.jsp";

public String m_configLocation = "/oracle";
public static final String ms_protocol = "http";
ResourceRequest rrq = new ResourceRequest(ms_protocol, ms_resource,
                                          ms_method);

UserSession session = new UserSession(sessionToken)
if(session.isAuthorized(rrq))
    return true;
else
    return false;
```

Useful resources:

http://docs.oracle.com/cd/E23943_01/apirefs.1111/e22472/toc.htm
http://docs.oracle.com/cd/E23943_01/doc.1111/e12491/as_api.htm

Displaying different views with different/relevant data to users will be done at frontend.

6.3 Data Privacy

There are no specific data privacy issues. No encryption or hashing of data is necessary in the data base. The username, password and other information is stored in the Oracle Access Manager (OAM).

The access to the site itself will be done using HTTPS. This will be configured in the web server.

6.4 Data Integrity

User Input data needs to be validated before being passed to the back-end services. No specific data integrity issues exist. All save operations must be thread-safe and transactional.

6.5 Data Migration

None.

7 Deployment and Development Concerns

7.1 Logical Deployment

The application can be deployable in JBoss Application Server. The Frontend tier will incorporate HTMLs with JSP. The application will be deployed as a single war file. The services will also be deployed on a JBoss AS, and use Hibernate to access the database. Spring is used to give necessary configuration.

7.2 Building an Enterprise Service Bus using JBoss ESB

An ESB offers capabilities to connect new and existing software applications within and across enterprises, with a rich set of features enabling management and monitoring of interactions between applications

ESB offers the following features

- First, is the ability for the ESB to interact with service requesters and service providers, in a variety of ways: by sending to and receiving messages from a persistent messaging backbone, particularly MQ, and being able to send and receive Web services request and response messages over HTTP and JMS, with support for standards like the Web Services Interoperability (WS-I) Basic Profile 1.1. We will focus initially on standards based messages using SOAP and XML, though supporting other message formats such as text and binary are important, too.
- Next, is the ability to convert between different message and transport protocols, such as SOAP over HTTP to SOAP over JMS.
- Then, we want to be able to transform XML messages using the popular transformation language XSLT.
- Another basic feature is to be able to apply message mediations such as logging. Moreover, we very frequently have some type of context-based dynamic routing taking place.

Useful Resources:

http://docs.jboss.org/jbossesb/docs/4.11/manuals/html/Getting_Started_Guide/index.html

http://docs.jboss.org/jbossesb/docs/4.11/manuals/html/Services_Guide/index.html

http://docs.jboss.org/jbossesb/docs/4.11/manuals/html/Programmers_Guide/index.html

7.3 Metrics Requirements

- The application does not propose any limits.

7.3.1 Page Performance Requirements

- The portal will have 24x7x365 uptime with reasonably high level of availability.
- All pages in the application should be loaded in 3 seconds or less seconds or less after the initial hit.
- At its peak usage, the number of concurrent users should not exceed 100.
- The average number of users on the system at any one time is anticipated to be in between 25-30.

7.4 Technologies

7.4.1 Software

- Oracle RAC 11g
- J2EE 6.0
- JAVA SDK 5.0.22
- Spring 3.0.5

7.4.2 Database

- Oracle RAC 11g

7.4.3 Web Server

7.4.4 Application Server

- JBoss 5.1

7.4.5 Frameworks

- Spring MVC
- HornetQ 2.2 <https://www.jboss.org/hornetq/>
- JBoss ESB 4.11 <http://www.jboss.org/jbossesb>
- Riftsaw 2.3.0 <http://www.jboss.org/riftsaw>
- Hibernate 3.6.3
- Quartz - <http://quartz-scheduler.org/>
- Drools - <http://www.jboss.org/drools>

7.4.6 Unit Testing

- JUnit-5

7.4.7 Operating System

- CentOS 6

The application should support all latest versions of OS type as the application can be used by other US states.

7.4.8 TC Components

- PDF Builder 1.0

7.4.9 Web technology

- HTML/CSS
- IE6/IE7/IE8 - latest version
- FireFox - latest version
- Safari - latest version
- Chrome - latest version

7.5 User Interface Implementation

7.5.1 Prototype conversion

The application will use JSTL.

7.5.2 Resolution

The users can access the application though various computer configurations with different screen sizes, based on the specific client capabilities.

.