(1) Brief description of the physical architecture

* Multitier architecture: have more than one client applications and more than one logic/data tier
* Resource pooling (more than one client share common pool)
* Security (password hashing used)
* Service-orientated architecture: offers web services
* Component-based architecture: component communication by use of interfaces

(2) You are required to provide instructions for clinics to deploy and test your system with sample data.

* Clean and build all project files first
* Connect a database to the project
* Deploy ejb
* Run IS2103Assignment2CatClient on your administrative terminals
* Run IS2103Assignment2KioskClient on your patient-facing kiosks
* IS2103Assignment2AMSClient is a demonstration as to how a client remotely accessing your
* services would interact with them.

(3) Document the business rules, rationales and assumptions for your logical data model

**ENTITIES**

In our implementation of CARS we have noted 6 important entity classes to include.

|  |  |
| --- | --- |
| **Entity** | **Description** |
| AppointmentEntity | Encapsulate information about Appointments & Consultations |
| ClinicEntity | Encapsulate information about the clinic itself |
| DoctorEntity | Encapsulate information about Doctor |
| LeaveEntity | Encapsulate information about leave(s) of Doctors |
| PatientEntity | Encapsulate information about Patient |
| StaffEntity | Encapsulate information about Staff |

**Appointment Entity**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Data Type** | **Constraints** | **Description** |
| appointmentId | Long | * Identity * Auto-generated Value | Unique Id of Appointment |
| patient | PatientEntity | * Many to one * False if null | Patient associated with appointment |
| doctor | DoctorEntity | * Many to one * False if null | Doctor associated with appointment |
| appointmentTime | LocalDateTime | * False if null | Date and time of appointment |
| dateTime | String | * False if null | String copy of appointmentTime for marshalling |
| isCancelled | Boolean | - | Check for if appointment has been cancelled |
| isConfirmed | Boolean | - | Check for if appointment has been confirmed |

|  |  |  |
| --- | --- | --- |
| **Method** | **Return Type** | **Description** |
| cancelAppointment() | Void | Sets isCancelled Boolean to true |
| getAppointmentId() | Long | Returns appointment Id |
| getPatient() | PatientEntity | Returns patient entity of this appointment |
| setPatient(PatientEntity pe) | Void | Sets patient entity of this appointment to the value in parenthesis |
| getDoctor() | DoctorEntity | Returns doctor entity of this appointment |
| setDoctor(DoctorEntity de) | Void | Sets doctor entity of this appointment to the value in parenthesis |
| getPatientId() | Long | Returns patient Id of patient entity of this appointment |
| getDoctorId() | Long | Returns doctor Id of doctor entity of this appointment |
| getAppointmentTime() | LocalDateTime | Returns appointment time in LocalDateTime type |
| setAppointmentTime(LocalDateTime appointmentTime) | Void | Sets appointmentTime of this appointment to the value in parenthesis |
| getIsCancelled() | Boolean | Returns true if appointment has been cancelled already, else it returns false. |
| getIsConfirmed() | Boolean | Returns true if appointment has been confirmed already, else it returns false. |
| setIsConfirmed() | Void | Sets value of isConfirmed to true to confirm appointment |
| getDateTime() | String | Returns string value of dateTime of appointment |
| setDateTime(String dateTime) | Void | Sets dateTime of this appointment to the value in parenthesis |

**Clinic Entity**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Data Type** | **Constraints** | **Description** |
| id | Long | * Identity * Auto-generated Value | Unique Id of Clinic Entity |
| operationDay | DayOfWeek | * Enum String | Day of the week at the clinic |
| startOfDay | LocalTime | * Many to one * False if null | Starting time of that day at clinic |
| endOfDay | LocalTime | * False if null | Closing time of the day at clinic |

|  |  |  |
| --- | --- | --- |
| **Method** | **Return Type** | **Description** |
| getId() | Long | Returns clinic id |
| setId(Long id) | Void | Set clinic id to value in parenthesis |
| getDay() | DayOfWeek | Return operationDay value |
| setDay(DayOfWeek day) | Void | Sets operationDay to value in parenthesis |
| getStartOfDay() | LocalTime | Returns time of start of the operationDay |
| setStartOfDay(LocalTime startOfDay) | Void | Sets start of operation day timing to value in parenthesis |
| getEndOfDay() | LocalTime | Returns time of end of the operationDay |
| setEndOfDay(LocalTime endOfDay) | void | Sets end of operation day timing to value in parenthesis |

**Doctor Entity**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Data Type** | **Constraints** | **Description** |
| doctorId | Long | Identity, Auto-generated Value | Unique Id of Doctor |
| firstName | String | **-** | First name of doctor |
| lastName | String | **-** | Last name of doctor |
| Registration | String | **-** | Doctor’s registration |
| qualification | String | **-** | Doctor’s qualification |

|  |  |  |
| --- | --- | --- |
| **Method** | **Return Type** | **Description** |
| getDoctorId() | Long | Returns doctorId value |
| setDoctorID(Long doctorId) | Void | Takes in a Long value to replace current doctored value |
| getFirstName() | String | Returns first name of doctor |
| setFirstName(String firstName) | Void | Sets firstName of doctor to the string in parenthesis |
| getLastName() | String | Return last name of doctor |
| setLastName(String lastName) | Void | Sets lastName of doctor to the string in parenthesis |
| getFullName() | String | Returns the full name of the doctor (firstName+” “+lastName) |
| getRegistration() | String | Returns registration of doctor |
| setRegistration(String registration) | Void | Sets registration of doctor to value of string in parenthesis |
| getQualification() | String | Returns qualification of doctor |
| setQualification(String qualification) | void | Sets qualification of doctor to value of string in parenthesis |

**Leave Entity**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Data Type** | **Constraints** | **Description** |
| leaveId | Long | * Identity * Auto-generated Value | Unique Id of leave record |
| doctor | DoctorEntity | * Many to one * False if null | Doctor Entity associated with doctor of this leave record |
| leaveDate | LocalDate | * False if null | Date of leave |

|  |  |  |
| --- | --- | --- |
| **Method** | **Return Type** | **Description** |
| getLeaveId() | Long | Returns id of this Leave |
| setLeaveId(Long leaveId) | Void | Sets id of this leave to value in parenthesis |
| getDoctorId() | Long | Returns id of doctor associated with this leave |
| getDoctorEntity() | DoctorEntity | Returns Doctor Entity of doctor associated with this leave |
| getLeaveDate() | LocalDate | Return date of leave |
| setLeaveDate(LocalDate leaveDate) | Void | Sets date of this leave to value in parenthesis |

**Patient Entity**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Data Type** | **Constraints** | **Description** |
| patientId | Long | * Identity * Auto-generated Value | Unique Id of patient |
| identityNumber | String | * Unique = true * False if null * Length = 9 | Unique identity number for patient |
| firstName | String | * False if null | Patient’s first name |
| lastName | String | * False if null | Patient’s last name |
| Gender | String | * False if null | Patient’s gender |
| Age | Integer | * False if null | Patient’s age |
| Phone | String | * False if null | Patient’s phone number |
| Address | String | * False if null | Patient’s address |
| password | String | * False if null | Patient’s password |

|  |  |  |
| --- | --- | --- |
| **Method** | **Return Type** | **Description** |
| getPatientId() | Long | Returns id of this patient |
| setPatientId(Long patientId) | Void | Sets id of this patient to value in parenthesis |
| getIdentityNumber() | String | Returns identity number of patient |
| setIdentityNumber(String identityNumber) | Void | Sets identity number of this patient to value in parenthesis |
| getFirstName() | String | Return patient’s first name |
| setFirstName(String firstName) | Void | Sets first name of this patient to value in parenthesis |
| getLastName() | String | Return last name of patient |
| setLastName(String lastName) | Void | Sets last name of this patient to value in parenthesis |
| getGender() | String | Return gender of patient |
| getAge() | Integer | Return age of patient |
| setAge(Integer age) | Void | Sets age of this patient to value in parenthesis |
| getPhone() | String | Return phone number of patient |
| setPhone(String phone) | Void | Sets phone number of this patient to value in parenthesis |
| getAddress() | String | Return address of patient |
| setAddress(String address) | Void | Sets address of this patient to value in parenthesis |
| getPassword() | String | Return password of patient |
| setPassword(String password) | Void | Sets password of this patient to value in parenthesis |

**Staff Entity**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Data Type** | **Constraints** | **Description** |
| staffId | Long | * Identity * Auto-generated Value | Unique Id of staff |
| firstName | String | * Identity * Auto-generated Value | Staff’s first name |
| lastName | String | * Identity * Auto-generated Value | Staff’s last name |
| userName | String | * False if null | Staff’s username |
| password | String | * False if null | Staff’s password |

|  |  |  |
| --- | --- | --- |
| **Method** | **Return Type** | **Description** |
| getStaffId() | Long | Returns id of this staff |
| setStaffId(Long staffId) | Void | Sets id of this staff to value in parenthesis |
| getFirstName() | String | Return staff’s first name |
| setFirstName(String firstName) | Void | Sets first name of this staff to value in parenthesis |
| getLastName() | String | Return last name of staff |
| setLastName(String lastName) | Void | Sets last name of this staff to value in parenthesis |
| getUserName() | String | Return username of staff |
| setUserName(String userName) | Integer | Sets username of this staff to value in parenthesis |
| getPassword() | String | Return password of staff |
| setPassword(String password) | Void | Sets password of this staff to value in parenthesis |

**UTIL.EXCEPTION**

|  |  |
| --- | --- |
| Exception | Description |
| AppointmentAlreadyCancelledException | Occurs when appointment being cancelled has been cancelled already |
| AppointmentNotFoundException | Occurs when retrieving an appointment that does not exist in database |
| DoctorNotFoundException | Occurs when retrieving a doctor that does not exist in database |
| EntityInstanceExistsInCollectionException | Occurs when creating an entity that already exists in database |
| EntityManagerException |  |
| EntityMismatchException | Occurs when entity created to delete from the database does not match with any existing entity |
| InvalidLoginCrednetialException | Occurs when login credentials are not existent in database |
| LeaveNotFoundException | Occurs when retrieving a leave that does not exist in database |
| PatientNotFoundException | Occurs when retrieving a patient that does not exist in database |
| StaffNotFoundException | Occurs when retrieving a staff that does not exist in database |

**KIOSK CLIENT**

|  |  |
| --- | --- |
| Class | Description |
| Is2103AssignmentKioskClient | Calls method runApp in MainApp to run business operations |
| MainApp | Stores all business operations to occur in Self-Service Kiosks in the clinic |

**Is2103AssignmentKioskClient**

|  |  |  |
| --- | --- | --- |
| Attributes | Data Type | Description |
| apointmentEntitySessionBeanRemote | ApointmentEntitySessionBeanRemote | To call CRUD operations from session bean for appointment entities |
| doctorEntitySessionBeanRemote | DoctorEntitySessionBeanRemote | To call CRUD operations from session bean for doctor entities |
| patientEntitySessionBeanRemote | PatientEntitySessionBeanRemote | To call CRUD operations from session bean for patient entities |
| queueGeneratorSessionBeanRemote | QueueGeneratorSessionBeanRemote | To call CRUD operations from session bean for queue entities |
| leaveEntitySessionBeanRemote | LeaveEntitySessionBeanRemote | To call CRUD operations from session bean for leave entities |

**MainApp**

|  |  |  |
| --- | --- | --- |
| Attributes | Data Type | Description |
| apointmentEntitySessionBeanRemote | ApointmentEntitySessionBeanRemote | To call CRUD operations from session bean for appointment entities |
| doctorEntitySessionBeanRemote | DoctorEntitySessionBeanRemote | To call CRUD operations from session bean for doctor entities |
| patientEntitySessionBeanRemote | PatientEntitySessionBeanRemote | To call CRUD operations from session bean for patient entities |
| queueGeneratorSessionBeanRemote | QueueGeneratorSessionBeanRemote | To call CRUD operations from session bean for queue entities |
| leaveEntitySessionBeanRemote | LeaveEntitySessionBeanRemote | To call CRUD operations from session bean for leave entities |
| currentPatientEntity | PatientEntity | Holds reference to the patient entity of the patient currently logged-in |

**MainApp Methods:**

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Return Type | Description | Rules, Rationale, Assumptions |
| doRegister() | Void | Registers an account for patient | * Will create account as long as data entered is follows attribute constraints of PatientEntity |
| doLogin() | void | Logins in patient on the kiosk | * Will log patient in as long as entered username and password exists in a patient Entity in the database |
| menuMain() | Void | Displays main menu of operations and allows user to choose operations to perform |  |
| doRegisterWalkIn() | Void | Registers a walk-in consultation for logged in patient | * Can only register for the next 3 hours and within operation hours * Must be registered for consultation with doctor who is available and does not have another appointment at the same time |
| doRegisterAppointment() | Void | Confirms a pre-booked appointment for logged in patient | * Can confirm appointment if appointment associated with appointment Id entered is associated with an appointment that is not cancelled or already confirmed |
| doViewAppointment() | Void | Views all appointments made under the patient’s account | * Displays all existing appointments under logged-in patient in database |
| doAddAppointment() | Void | Adds a new appointment under the patient’s account | * Appointment cannot be with a doctor that is unavailable (on leave or has another appointment) * Appointment must be within operation hours * Appointment must be booked at least 2 days in advance |
| doCancelAppointment() | void | Cancels an existing appointment under the patient’s account | * Appointment must not be already cancelled * Appointment must exist in database |

**CATS CLIENT**

|  |  |
| --- | --- |
| Class | Description |
| AdministrationOperationModule | Stores all business logic to occur in Administration Operation section of CARS |
| AppointmentOperationModule | Stores all business logic to occur in Appointment Operation section of CARS |
| Main | Calls method runApp in MainApp to run business operations |
| MainApp | Displays main menu for log in and register. Can call any of the 2 OperationModule classes for further operation. |
| RegistrationOperationModule | Stores all business logic to occur in Registration Operation section of CARS |

**Main**

|  |  |  |
| --- | --- | --- |
| Attributes | Data Type | Description |
| apointmentEntitySessionBeanRemote | ApointmentEntitySessionBeanRemote | To call CRUD operations from session bean for appointment entities |
| doctorEntitySessionBeanRemote | DoctorEntitySessionBeanRemote | To call CRUD operations from session bean for doctor entities |
| patientEntitySessionBeanRemote | PatientEntitySessionBeanRemote | To call CRUD operations from session bean for patient entities |
| queueGeneratorSessionBeanRemote | QueueGeneratorSessionBeanRemote | To call CRUD operations from session bean for queue entities |
| leaveEntitySessionBeanRemote | LeaveEntitySessionBeanRemote | To call CRUD operations from session bean for leave entities |

**MainApp**

|  |  |  |
| --- | --- | --- |
| Attributes | Data Type | Description |
| apointmentEntitySessionBeanRemote | ApointmentEntitySessionBeanRemote | To call CRUD operations from session bean for appointment entities |
| doctorEntitySessionBeanRemote | DoctorEntitySessionBeanRemote | To call CRUD operations from session bean for doctor entities |
| patientEntitySessionBeanRemote | PatientEntitySessionBeanRemote | To call CRUD operations from session bean for patient entities |
| queueGeneratorSessionBeanRemote | QueueGeneratorSessionBeanRemote | To call CRUD operations from session bean for queue entities |
| leaveEntitySessionBeanRemote | LeaveEntitySessionBeanRemote | To call CRUD operations from session bean for leave entities |
| administrationOperationModule | AdministrationOperationModule | Reference to AdministrationOperationModule class |
| registrationOperationModule | RegistrationOperationModule | Reference to RegistrationOperationModule class |
| appointmentOperationModule | AppointmentOperationModule | Reference to AppointmentOperationModule class |
| currentStaffEntity | StaffEntity | Reference to logged-in staff |

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Return Type | Description | Rules, Rationale, Assumptions |
| doLogin() | Void | Login for staff | * Assigns staff to currentStaffEntity attribute for further operation to be made by this entity |
| menuMain() | Void | Displays main menu of operations and allows user to choose operations to perform |  |

**Administration Operation Module**

|  |  |  |
| --- | --- | --- |
| Attributes | Data Type | Description |
| apointmentEntitySessionBeanRemote | ApointmentEntitySessionBeanRemote | To call CRUD operations from session bean for appointment entities |
| doctorEntitySessionBeanRemote | DoctorEntitySessionBeanRemote | To call CRUD operations from session bean for doctor entities |
| patientEntitySessionBeanRemote | PatientEntitySessionBeanRemote | To call CRUD operations from session bean for patient entities |
| leaveEntitySessionBeanRemote | LeaveEntitySessionBeanRemote | To call CRUD operations from session bean for leave entities |

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Return Type | Description | Rules, Rationale, Assumptions |
| menuAdministrationOperation() | Void | Displays a menu and allows logged-in staff to select administration operations to carry out |  |
| doManagePatients() | void | Displays menu for Patient management, for operations regarding patient management |  |
| doManageDoctors() | Void | Displays menu for Doctor management, for operations regarding doctor management |  |
| doManageStaffs() | Void | Displays menu for staff management, for operations regarding staff management |  |
| doAddPatient() | Void | Adds new patient to database | * Input entered must be valid to create * Patient must not already exist in database |
| doViewPatient() | Void | Views patient’s details | * Patient must already exist in database |
| doUpdatePatient() | Void | Update patient details | * Patient must already exist in database * Identity number used to make copy of new patient to replace old copy must match the already existing entity |
| doDeletePatient() | void | Delete existing patient from database | * Patient must already exist in database |
| doViewAllPatients() | Void | View list of all patient details |  |
| doAddDoctor() | void | Create new doctor in database | * Input entered must be valid to create * Doctor must not already exist in database |
| doViewDoctor() | void | View doctor details | * Doctor must already exist in database |
| doUpdateDoctor() | void | Update doctor details | * doctor must already exist in database * doctorId used to make copy of new doctor to replace old copy must match the already existing entity |
| doDeleteDoctor() | void | Delete doctor from database | * Doctor must already exist in database |
| doViewAllDoctors | void | Show list of all doctor details |  |
| doAddStaff() | Void | Create new staff in database | * Input entered must be valid to create * staff must not already exist in database |
| doViewStaff() | Void | View staff details | * Staff must already exist in database |
| doUpdateStaff() | Void | Update staff details | * staff must already exist in database * staffId used to make copy of new doctor to replace old copy must match the already existing entity |
| doDeleteStaff() | Void | Delete staff from database | * staff must already exist in database |
| doViewAllStaff() | void | View list of all staff details |  |
| doManageLeaves() | Void | Go to leave management section |  |
| doAddLeave() | void | Adds leave for a doctor | * Leave must be made a week in advance * Only one leave per week * Doctor associated to doctored entered must exist in database |

**Appointment Operation Module**

|  |  |  |
| --- | --- | --- |
| Attributes | Data Type | Description |
| apointmentEntitySessionBeanRemote | ApointmentEntitySessionBeanRemote | To call CRUD operations from session bean for appointment entities |
| doctorEntitySessionBeanRemote | DoctorEntitySessionBeanRemote | To call CRUD operations from session bean for doctor entities |
| patientEntitySessionBeanRemote | PatientEntitySessionBeanRemote | To call CRUD operations from session bean for patient entities |

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Return Type | Description | Rules, Rationale, Assumptions |
| menuAppointmentOperation() | Void | Displays a menu and allows logged-in staff to select administration operations to carry out |  |
| viewAppointments() | void | View all logged-in patient’s appointments | * Patient must be existent in database |
| addAppointment() | Void | Add appointment for logged-in patient | * Appointments must be booked at least 2 days in advance * Appointment must be made with available doctor at an available timing |
| cancelAppointment() | Void | Cancel appointment for logged-in patient | * Appointment must already exist in database * Appointment must not already be cancelled |

**Registration Operation Module**

|  |  |  |
| --- | --- | --- |
| Attributes | Data Type | Description |
| apointmentEntitySessionBeanRemote | ApointmentEntitySessionBeanRemote | Reference to session bean for appointment entities |
| doctorEntitySessionBeanRemote | DoctorEntitySessionBeanRemote | Reference to session bean for doctor entities |
| patientEntitySessionBeanRemote | PatientEntitySessionBeanRemote | Reference to session bean for patient entities |
| leaveEntitySessionBeanRemote | LeaveEntitySessionBeanRemote | Reference to session bean for leave entities |
| queueEntitySessionBeanRemote | QueueEntitySessionBeanRemote | Reference to session bean for queue entities |

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Return Type | Description | Rules, Rationale, Assumptions |
| menuRegistrationOperation() | Void | Displays a menu and allows logged-in staff to select administration operations to carry out |  |
| registerNewPatient() | void | Create new patient in database | * Input entered must be valid to create * Patient must not already exist in database |
| registerWalkInConsult() | Void | Registers a walk-in consultation for logged in patient | * Can only register for the next 3 hours and within operation hours * Must be registered for consultation with doctor who is available and does not have another appointment at the same time |
| registerAppointmentConsult() | Void | Confirms a pre-booked appointment for logged in patient | * Can confirm appointment if appointment associated with appointment Id entered is associated with an appointment that is not cancelled or already confirmed |

**AMS CLIENT**

|  |  |
| --- | --- |
| Class | Description |
| Main | Calls method runApp in MainApp to run business operations |
| MainApp | Displays main menu for log in and register. Can call any of the 2 OperationModule classes for further operation. |

**Main**

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Return Type | Description | Rules, Rationale, Assumptions |
| main(String[] args) | Void | Main method calls mainApp to run application |  |

**MainApp**

|  |  |  |
| --- | --- | --- |
| Attributes | Data Type | Description |
| currentPatient | PatientEntity | Reference to logged-in patient |

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Return Type | Description | Rules, Rationale, Assumptions |
| doRegister() | Void | Creates new patient in database | * Patient must not already exist in database |
| doLogin() | Void | Login for staff | * Assigns staff to currentStaffEntity attribute for further operation to be made by this entity |
| menuMain() | Void | Displays main menu of operations and allows user to choose operations to perform |  |
| doViewAppointments() | Void | View all of patient’s appointments | * Patient must be existent in database |
| doAddAppointment() | Void | Add appointment for patient | * Appointments must be booked at least 2 days in advance * Appointment must be made with available doctor at an available timing |
| doCancelAppointment() | void | Cancel appointment for patient | * Appointment must already exist in database * Appointment must not already be cancelled |

**Web methods for AMSClient:**

1. java.util.List<ws.client.ams.DoctorEntity> retrieveDoctorsAvailableOnDate(java.lang.String now)
2. java.util.List<java.lang.String> retrieveAppointmentByPatientIdentityNo(java.lang.String identityNumber)
3. ws.client.ams.PatientEntity patientLogin(java.lang.String identityNumber, java.lang.String password)
4. ws.client.ams.PatientEntity createNewPatient(java.lang.String identityNumber, java.lang.String password, java.lang.String firstName, java.lang.String lastName, java.lang.String gender, java.lang.Integer age, java.lang.String phone, java.lang.String address)

These methods were used in AMSClient in place of session bean CRUD methods.

**IS2103Assignment2-ejb**

|  |  |
| --- | --- |
| Package | Session Beans |
| Ejb.session.singleton | * DataInitializationSessionBean * QueueGeneratoeSessionBean * QueueGenerationSessionBeanLocal |
| Ejb.session.stateless | * AppointmentEntityController * AppointmentEntityControllerLocal * AppointmentEntitySessionBean * AppointmentEntitySessionBeanLocal * ClinicEntitySessionBean * ClinicEntitySessionBeanLocal * DoctorEntityController * DoctorEntityControllerLocal * DoctorEntitySessionBean * DoctorEntitySessionBeanLocal * LeaveEntitySessionBean * LeaveEntitySessionBeanLocal * PatientEntityController * PatientEntityControllerLocal * PatientEntitySessionBean * PatientEntitySessionBeanLocal * StaffEntitySessionBean * StaffEntitySessionBeanLocal |
| Ejb.session.ws | * IS2103Assignment2WebService |

**Data Initialization Session Bean**

|  |  |  |
| --- | --- | --- |
| Attributes | Data Type | Description |
| appointmentEntitySessionBeanLocal | AppointmentEntitySessionBeanLocal | To call CRUD operations from session bean for appointment entities |
| doctorEntitySessionBeanLocal | DoctorEntitySessionBeanLocal | To call CRUD operations from session bean for doctor entities |
| patientEntitySessionBeanLocal | PatientEntitySessionBeanLocal | To call CRUD operations from session bean for patient entities |
| clinicEntitySessionBeanLocal | ClinicEntitySessionBeanLocal | To call CRUD operations from session bean for clinic entities |

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Return Type | Description | Rules, Rationale, Assumptions |
| intializeData() | Void | Initializes sample database when ejb is deployed | * Entity will not be created if it already exists in database |

**Queue Generator Session Bean**

* QueueGeneratorSessionBeanLocal is a superclass of this class

|  |  |  |
| --- | --- | --- |
| Attributes | Data Type | Description |
| queueNumber | Integer | Queue number to be issued to patients for their consultation |

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Return Type | Description | Rules, Rationale, Assumptions |
| getNextQueueNumber() | Integer | Returns queue number | * Singleton session bean will just keep incrementing value of queue Number until lifecycle |

**Appointment Entity Session Bean**

* AppointmentEntitySessionBeanLocal , AppointmentEntitySessionBeanRemote are superclasses of this class
* AppointmentEntityController, AppointmentEntityControllerLocal have the same logic as their session bean counterparts

|  |  |  |
| --- | --- | --- |
| Attributes | Data Type | Description |
| entityManager | EntityManager  \*PersistenceContext annotation used here | Reference to entity manager to use java persistence |

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Return Type | Description | Rules, Rationale, Assumptions |
| createNewAppointment() | Long | Creates new appointment and returns appointment Id |  |
| retrieveAllAppointments() | List<AppointmentEntity> | Return list of appointments |  |
| retrieveAppointmentByAppointmentId(Long appointmentId) | AppointmentEntity | Returns appointment associated with appointmentId provided | * Appointment must exist in database |
| retrieveAppointmentByPatientIdentityNo(String patientId) | List<String> | Returns a list of appointments of the patient associated with the identity number provided |  |
| retrieveDoctorAvailableSlotsOnDay(DoctorEntity doctorEntity, LocalDate date) | List<LocalTime> | Provide a list of times a doctor is free for consultation on a certain day | * Timings should be within operation hours |
| hasAppointment(DoctorEntity doctorEntity, LocalDateTime appointmentTime) | String | Returns “X” if doctor is not free at the appointmentTime, else returns “O” | * To populate schedule for walk-in consults |
| confirmAppointment(Long patientId,Long appointmentId) | AppointmentEntity | Returns confirmed appointment | * Appointment must exist in database |
| cancelAppointment(AppointmentEntity appointmentEntity) | void | Cancels an existing appointment under the patient’s account | * Appointment must not be already cancelled * Appointment must exist in database |

**Clinic Entity Session Bean**

* ClinicEntitySessionBeanLocal , ClinicEntitySessionBeanRemote are superclasses of this class.

|  |  |  |
| --- | --- | --- |
| Attributes | Data Type | Description |
| entityManager | EntityManager  \*PersistenceContext annotation used here | Reference to entity manager to use java persistence |

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Return Type | Description | Rules, Rationale, Assumptions |
| createNewClinicTiming(ClinicEntity newClinicEntity) | Long | Creates new clinic timing and returns clinic Id |  |
| retrieveClinicTimingsByDay(String day) | List<ClinicEntity> | Return list of clinic timings |  |
| retrieveConsultationSlots(LocalDate seacrchDate) | List<LocalTime> | Returns list of consulation slots for the date entered | * Each day has different operation hours |

**Doctor Entity Session Bean**

* DoctorEntitySessionBeanLocal , DoctorEntitySessionBeanRemote are superclasses of this class
* DoctorEntityController, DoctorEntityControllerLocal have the same logic as their session bean counterparts

|  |  |  |
| --- | --- | --- |
| Attributes | Data Type | Description |
| entityManager | EntityManager  \*PersistenceContext annotation used here | Reference to entity manager to use java persistence |

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Return Type | Description | Rules, Rationale, Assumptions |
| createNewDoctor(DoctorEntity newDoctorEntity) | Long | Creates new doctor and returns doctor Id |  |
| retrieveAllDoctors() | List< DoctorEntity > | Return list of doctors |  |
| retrieveDoctorByDoctorId(Long doctorId) | DoctorEntity | Returns doctor associated with doctor id provided | * doctor must exist in database |
| retrieveDoctorByDoctorName(String firstName, String lastName) | DoctorEntity | Returns doctor associated with first name and last name provided | * doctor must exist in database |
| retrieveDoctorsAvailableOnDate(LocalDate date) | List< DoctorEntity > | Return list of doctors not on leave on the date entered |  |
| updateDoctor(DoctorEntity doctorEntity) | void | Updates doctor’s details |  |
| deleteDoctor(Long doctored) | void | Deletes doctor from database | * doctor must already exist in database |

**Leave Entity Session Bean**

* LeaveEntitySessionBeanLocal , LeaveEntitySessionBeanRemote are superclasses of this class

|  |  |  |
| --- | --- | --- |
| Attributes | Data Type | Description |
| entityManager | EntityManager  \*PersistenceContext annotation used here | Reference to entity manager to use java persistence |

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Return Type | Description | Rules, Rationale, Assumptions |
| createNewLeave(LeaveEntity newLeaveEntity) | Long | Creates new leave and returns leave Id |  |
| retrieveAllLeaves() | List< LeaveEntity > | Return list of all leaveEntity |  |
| retrieveLeaveByDoctorId(Long doctorId) | List< LeaveEntity > | Returns leaves of doctor associated with doctorId input | * doctor of doctorId must exist in database |
| retrieveLeaveByLeaveId(Long leaveId) | LeaveEntity | Returns leave associated with leaveId entered | * leave of leaveId must exist in database |
| retrieveLeaveByDate(LocalDate date) | List< LeaveEntity > | Return list of leaves for the date given |  |
| cancelLeave(LeaveEntity leaveEntity) | void | Cancels leave |  |

**Patient Entity Session Bean**

* PatientEntitySessionBeanLocal , PatientEntitySessionBeanRemote are superclasses of this class
* PatientEntityController, PatientEntityControllerLocal have the same logic as their session bean counterparts

|  |  |  |
| --- | --- | --- |
| Attributes | Data Type | Description |
| entityManager | EntityManager  \*PersistenceContext annotation used here | Reference to entity manager to use java persistence |

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Return Type | Description | Rules, Rationale, Assumptions |
| createNewPatient(PatientEnity newPatientEntity) | Long | Creates new patient and returns patient Id | * Patient cannot already exist in database |
| patientLogin(String identityNumber, String password) | PatientEntity | Logs patient in | * patient must already exist in database * login credentials must be correct |
| retrieveAllPatients() | List< PatientEntity > | Returns list of patients |  |
| retrievePatientByPatientId(Long patientId) | PatientEntity | Returns patient associated with patient id entered | * patient must already exist in database |
| retrievePatientByPatientIdentityNumber(String identityNumber) | PatientEntity | Returns patient associated with patient identity number entered | * patient must already exist in database |
| updatePatient(PatientEntity patientEntity) | void | Update patient details | * patient must already exist in database |
| deletePatient(Long patientId) | void | Delete patient | * patient must already exist in database |
| hashPassword(String password) | String | Returns encrypted version of entered password | * for security reasons |

**Staff Entity Session Bean**

* StaffEntitySessionBeanLocal , StaffEntitySessionBeanRemote are superclasses of this class

|  |  |  |
| --- | --- | --- |
| Attributes | Data Type | Description |
| entityManager | EntityManager  \*PersistenceContext annotation used here | Reference to entity manager to use java persistence |

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Return Type | Description | Rules, Rationale, Assumptions |
| createNewStaff(StaffEntity newStaffEntity) | Long | Creates new staff and returns staff Id | * staff cannot already exist in database |
| StaffEntity staffLogin(String username, String password) | StaffEntity | Logs staff in | * staff must already exist in database * login credentials must be correct |
| retrieveAllStaffs() | List< StaffEntity > | Returns list of staff |  |
| retrieveStaffByUsername(String username) | StaffEntity | Returns staff associated with staff username entered | * staff must already exist in database |
| retrieveStaffByStaffId(Long staffId) | StaffEntity | Returns staff associated with staff id entered | * staff must already exist in database |
| updateStaff(StaffEntity staffEntity) | void | Update staff details | * staff must already exist in database |
| deleteStaff(Long staffId) | void | Delete staff | * staff must already exist in database |

**IS2103Assignment2WebService**

* contains web methods for ams client

|  |  |  |
| --- | --- | --- |
| Attributes | Data Type | Description |
| apointmentEntitySessionBeanRemote | ApointmentEntitySessionBeanRemote | Reference to session bean for appointment entities |
| doctorEntitySessionBeanRemote | DoctorEntitySessionBeanRemote | Reference to session bean for doctor entities |
| patientEntitySessionBeanRemote | PatientEntitySessionBeanRemote | Reference to session bean for patient entities |

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Return Type | Description | Rules, Rationale, Assumptions |
| createNewPatient | PatientEntity | Creates new patient and returns entity | * patient cannot already exist in database |
| createNewAppointment | AppointmentEntity | Creates new appointment and returns entity | * appointment cannot already exist in database |
| patientLogin | PatientEntity | Log patient in | * patient must exist in database * correct credentials entered |
| retrieveAppointmentByPatientIdentityNo | List<String> | Returns appointment associated with identity Number entered | * appointment must already exist in database |
| retrieveDoctorsAvailableOnDate | List<DoctorEntity> | Returns list of available doctors on a certain date |  |
| retrieveDoctorByDoctorId | DoctorEntity | Return doctor or doctorId entered | * doctor must already exist in database |
| retrieveDoctorAvailableSlotsOnDay | List<String> | Return list of available timings for a doctor |  |
| cancelAppointment | AppointmentEntity | Cancels appointment and returns it | * appointment must already exist in database |