Provide a detailed explanation of how validation works in the OpenHDS. How are errors communicated between the different layers?

In the OpenHDS, violations conveyed between the different layer using following classes /methods:

Use of class ConstraintViolations:
OpenHDS uses the class ConstraintsViolations (extended from the class java.lang.Exception) to hold the error detected during the validation process. The appropriate messages are added using add(msg) method. The ConstraintsViolations object is injected as a parameter to the method setReferencedField(..) in the controller class. FieldBuilder.referenceField() method get called within this method. These overloaded methods in the FieldBuilder class check for the null violations. The other exceptions such as invalid fieldworker, individual are validated against the previously recorded data by calling the appropriate finder methods such. The violations /exceptions are added in the ConstraintsViolations cv using the method cv.addViolations(). The violations are later conveyed between different layers using the reference to this object. The AbstarctResource class checks for the violation using the cv.hasViolations() method. The HttpStatus is updated as bad request if any violations occurred. Then the response is sent back to controller class.

Use of javax.ConstraintValidator:
This is the second level validation process for validating the domain data before saving it to the database. The Openhds system has many custom ConstraintValidators which implements javax.ConstraintValidator interface. For example, while creating the membership , the start date is validated using the class CheckStartDateGreaterThanBirthDate. These custom validators make a call to isValid() method to check the domain data against some predefined constraints. The various javax.Validation class methods get called to check whether the validation process is successful or not. The class EntityServiceImpl calls to the validateEntity(item) method within the create(entityItem) method to check the violations occurred during this isValid() call. The corresponding entity is created in the database if validation is successful.

Example**:** The validation process during the creation of membership entity is as follows
1. The post membership request get mapped to MembershipResource insert(Membership membership) method. This method calls to the createResource(membership) of the class AbstractResource.
2. AbstractResource class instantiate the ConstraintViolations cv object in order to check the violations. The cv is injected as parameter to the setReferenceFeild(item, cv) method. This method calls back to the MembershipResource setReferncedField() method. A call to fieldBuilder class's referenceField() is made to check null or invalid data for the Fieldworker, Individual and SocialGroup entities within this method. The violation occurred during this process are added during using the violation.addViolation("errormessage") method.
3. The control is return back to AbstractResource class createResource() method. The violations are checked using the method cv.hasViolations(). If the violation occurred during this process the response with httpstatus.badRequest sent back to the Membershipresource controller class.

4. If there are no violations occurred during, AbstractResource make a call to MembershipResource.saveResource(item).

5. The saveResource(item) method make a call to membershipservice.createMembership() method where the evaluateMembership() method get called. It checks for the duplicate membership and death of individual. ConstraintViolations object is used to hold violation occurred during this process.

6. If no violation occurred during the evaluateMembeship() call , the entityservice.create() method get called.

7.The EntityService.create() method calls to the method validateEntity(entityitem) using the EntityValidationService object. This method validates the data against the various custom validations such as CheckStartDateGreaterThanBirtDate. The membership object is created if the above process get successful. The ConstraintViolations are thrown for the method saveResource(membership item) if there are any violations.