Eco EHR History of Present Illnesses

Application Overview:

The ECO-EHR Application HPI Module allows users such as medical practitioners, medical students, and doctors to add specific medical data about the patient such as infectious diseases, psychiatric problems, neurological problems, eye problems, ear problems, etc.

The sequence flow of the back-end consists of 5 layers to communicate with the database and retrieve info:

1. Controller layer
2. Service layer
3. Service implementation layer
4. Repository Layer
5. Entity Layer

❖ Getting All Encounters By Encounter ID Module

This module is used to get past encounter information about the patient using encounter ID.

A) Controller layer

EncounterHistoryWebService.java

The class file is present in the spring boot location package name: com.cpa.ehr.web.rest.system

The method used within the class to get all encounters by encounter ID is the *getAllEncounterByEncounterID(Long encounterId)* method.

Input:

Name: encounterId

Data-type: Long

Output:

Name : encDTO

Data Type : ResponseEntity

```
@RestController
@RequestMapping("/api/rest/encounterHistory")
@CrossOrigin(origins = { "http://localhost:4300" })
public class EncounterHistoryWebService {
@GetMapping("/getAllEncounterByEncounterId")
     public ResponseEntity<List<EncounterHistoryRecordDTO>>
getAllEncounterByEncounterId(@RequestParam("encounterId") Long
encounterId) {
          List<EncounterHistoryRecordDTO> encDTO =
encHistoryService.retrieveAllEncounterByEncounterId(encounterI
d);
          return new ResponseEntity<>(encDTO, HttpStatus.OK);
     }
}
```

B) Service layer

EncounterHistoryService.java

This is an interface that contains only method names without their implementation.

The file is present in the spring boot location package name: com.cpa.ehr.service.system

The method used in the service level to create the patient medication is the *retrieveAllEncounterByEncounterId(encounterId)*.

The class EncounterHistoryRecordDTO.java is used to store the object as a DTO.

Input:

Name: encounterId

Date-type: Long

Output:

List<EncounterHistoryRecordDTO> historyRecordList

```
	public interface EncounterHistoryService {
	List<EncounterHistoryRecordDTO>
etrieveAllEncounterByEncounterId(Long encounterId);
}
```

C) Service implementation layer

EncounterHistoryServiceImpl.java

The class implements the interface that was described above in the Service layer, containing the concrete implementation of all the previous methods.

The file is present in the spring boot location package name:

com.cpa.ehr.service.system.impl

The *retrieveAllEncounterByEncounterId(Long encounter Id)* method is used to retrieve all encounter data.

Input:

Name: encounterId

Date-type: Long

Output:

List<EncounterHistoryRecordDTO> historyRecordList

```
@Service
public class EncounterHistoryServiceImpl implements
EncounterHistoryService {
@Override
public List<EncounterHistoryRecordDTO>
retrieveAllEncounterByEncounterId(Long encounterId) {
	try {
	List<Object> historyList =
	encHistoryRepo.getAllEncountersByEncounterId(encounterId)
	;
	}
	 return (historyList != null)
```

```
                        ?
encounterHistoryRecordMapper.entityListToEncounterHistoryRecor
dDTOList(historyRecordList)
                        : null;
     } catch (Exception e) {
     String username = SecurityUtils.getCurrentUserLogin();
     String exceptionString = "Error while retrieving all
questions by encounterId {} " + "{" + encounterId + "} \n"
                        + emailService.getStackTrace(e);

emailService.sendExceptionEmail(exceptionString,username);

LOG.error("Error while retrieving all questions by encounterId
{} ", e);
     }
          return Collections.emptyList();
     }
```

D) Repository layer

EncounterHistoryRepository.java

The class consists of the repository layer which has in built functionality like the save and getById methods.

The file is present in the spring boot location package name:

com.cpa.ehr.backend.dao.system

The *getAllEncountersByEncounterId(encounterId)* method is used for the queries of all encounter data

Input:

Name: encounterId

Date-type: Long

Output:

List<Object> getAllEncountersByEncounterId

```
@Repository
@Transactional
public interface EncounterHistoryRepository extends
JpaRepository<EncounterHistory,Long> {

     //Repository for getting all data..
```

```
	@Query(value = EncounterHistoryConstants.FIND_ALL,
nativeQuery = true)
	List<Object>
getAllEncountersByEncounterId(@Param("encounterId") Long
encounterId);
}
```

E) Entity Layer

EncounterHistoryRecordDTO.java

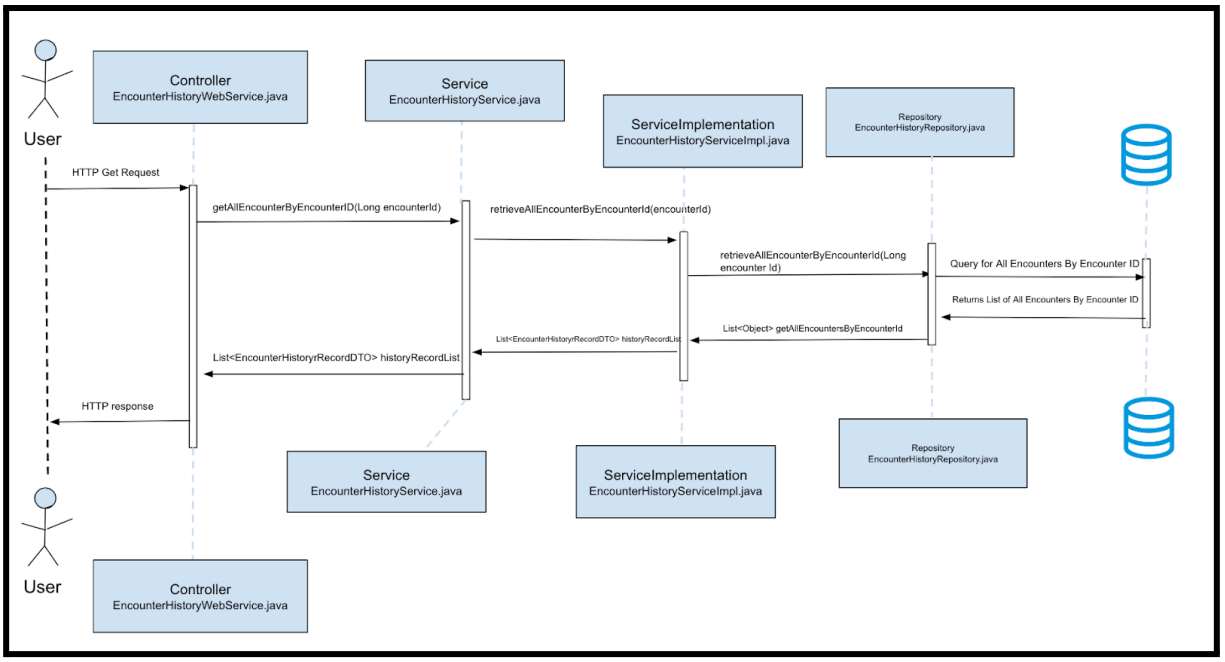
The file is present in the spring boot location package name:

com.cpa.ehr.backend.service.system.dto.

The layer describes the entity attributes of the encounter class.

```
public class EncounterHistoryRecordDTO {
private Long systemId;

	private String systemType;

	private String systemCode;

	private String systemDesc;

	private List<QuestionHistoryRecord> questionRecord;

	//Getters & Setters
}
```

Sequence Diagram:



❖ Creating Encounter Question Group (Multiple Checkboxes) Overview

This module is used to create an encounter question group when multiple checkboxes are checked by the user.

A) Controller layer

EncounterQuestionGroupWebService.java

The file is present in the spring boot location package name: com.cpa.ehr.web.rest.system

The method used within the class to insert the encounter question group is the *insertEncounterQuestionGroup(List <EncounterQuestionGroupDTO>)*

Input:

Name: <EncounterQuestionGroupDTO>

Date-type: List

Output:

Name : headers

Data Type : ResponseEntity

```
@RestController
@RequestMapping("/api/rest/encounterQuestionGroup")
@CrossOrigin(origins = { "http://localhost:4300" })
public class EncounterQuestionGroupWebService {
	@Autowired
	private EHRBaseService ehrBaseService;
	@Autowired
	private EncounterQuestionGroupService
encounterQuestionGroupService;

	@PostMapping("/createEncounterQuestionGroup")
	public ResponseEntity<EncounterQuestionGroupDTO>
insertEncounterQuestionGroup(@RequestBody
List<EncounterQuestionGroupDTO> encounterQuestionGroupDTO) {
		StaffMember loginUser =
ehrBaseService.currentUser();
		HttpHeaders headers= new HttpHeaders();
		for (EncounterQuestionGroupDTO
encQuestionGroupDTOSingleObj : encounterQuestionGroupDTO) {

FormatConverterUtils.setInitialDefaultValues(encQuestionGroupD
TOSingleObj, loginUser);

encounterQuestionGroupService.persistEncounterQuestionGroup(en
cQuestionGroupDTOSingleObj);
		}
		return new ResponseEntity<>(headers,
HttpStatus.CREATED);
	}
```

B) Service layer

EncounterQuestionGroupService.java

This is an interface that contains only method names without their implementation.

The file is present in the spring boot location package name: com.cpa.ehr.service.system

The *persistEncounterQuestionGroup(EncounterQuestionGroupDTO encQuestionGroupDTOSingleObj)* method call is done at the service level.

Input:

Name: encQuestionGroupDTOSingleObj

Date-type: EncounterQuestionGroupDTO

Output:

EncounterQuestionGroupDTO

```
public interface EncounterQuestionGroupService {

	EncounterQuestionGroupDTO
persistEncounterQuestionGroup(EncounterQuestionGroupDTO
encQuestionGroupDTOSingleObj);

}
```

C) Service Implementation layer

EncounterQuestionGroupServiceImpl.java

The class implements the interface that was described above in the Service layer, containing the concrete implementation of all the previous methods.

The logic is implemented in this class to persist the encounter quest group.

The file is present in the spring boot location package name:

com.cpa.ehr.service.system.impl

The *persistEncounterQuestionGroup(EncounterQuestionGroupDTO encounterQuestionGroupDTO)* method is used to persist the encounter question group.

Input:

Name: encounterQuestionGroupDTO

Date-type: EncounterQuestionGroupDTO

Output:

EncounterQuestionGroupDTO

```
@Service
public class EncounterQuestionGroupServiceImpl implements
EncounterQuestionGroupService {
	private static final Logger LOG =
LoggerFactory.getLogger(EncounterQuestionGroupServiceImpl.clas
s);
	@Autowired
	private EncounterQuestionGroupMapper
encounterQuestionGroupMapper;
	@Autowired
	private EncounterQuestionGroupRepository
encQuestionGroupRepository;
	@Autowired
	private EmailService emailService;
		@Override
	public EncounterQuestionGroupDTO
persistEncounterQuestionGroup(
			EncounterQuestionGroupDTO
encounterQuestionGroupDTO) {
		try {
			if (encounterQuestionGroupDTO != null) {
				EncounterQuestionGroup
newEncounterQuestionGroup = encounterQuestionGroupMapper

.encounterQuestionGroupDTOToEntity(encounterQuestionGroupDTO);

				//Spring boot for saving and retrieving
data save ,findAll, findById
				EncounterQuestionGroup
createdEncounterQuestionGroup = encQuestionGroupRepository

.save(newEncounterQuestionGroup);
				return (createdEncounterQuestionGroup !=
null)
```

```
                                    ?
encounterQuestionGroupMapper.entityToEncounterQuestionGroupDTO
(createdEncounterQuestionGroup)
                                    : null;
                    }
               } catch (Exception e) {
                    String username =
SecurityUtils.getCurrentUserLogin();
                    String exceptionString = "Error while
persistEncounterQuestionGroup {} " + "{" +
encounterQuestionGroupDTO
                              + "} \n" +
emailService.getStackTrace(e);

emailService.sendExceptionEmail(exceptionString, username);
               }
               return null;
          }
```

D) Repository layer

EncounterQuestionGroupRepository.java

The class consists of the repository layer which has inbuilt functionality like the save and getById method.

The inbuilt *save()* method is called for the query, saving, and retrieval of the data.

The file is present in the spring boot location package name:

com.cpa.ehr.backend.dao.system

No input

Output:

EncounterQuestionGroup

```
@Repository
@Transactional
public interface EncounterQuestionGroupRepository extends
JpaRepository<EncounterQuestionGroup, Long> {
      @Modifying
           @Query(value =
EncounterQuestionGroupConstants.DELETE_BY_ENCID, nativeQuery =
true)
          }
```

E) Entity Layer

encQuestionGroupDTOSingleObj.java

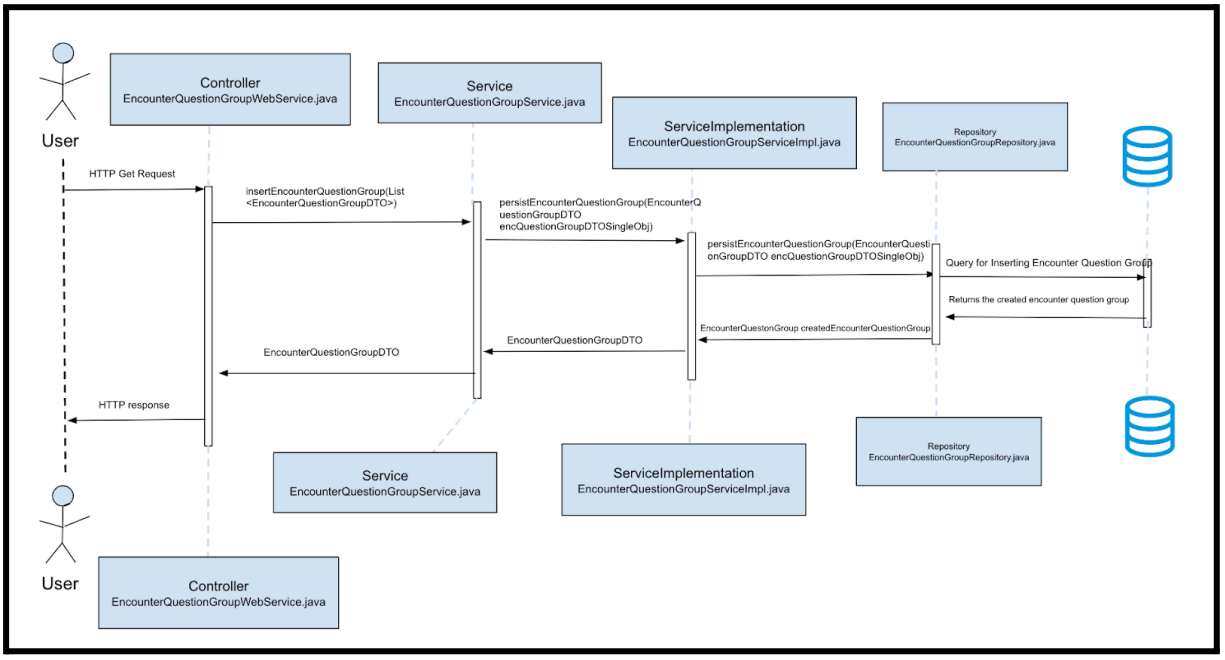
The file is present in the spring boot location package name:

com.cpa.ehr.backend.service.system.dto.

The layer describes the entity attributes of the encounter question group class.

```
package com.cpa.ehr.service.system.dto;
import java.util.Date;
public class EncounterQuestionGroupDTO {

	private Long encQuestionGroupId;

	private Long encounterId;

	private Long questionGroupId;

	private Long systemId;

	private String questionGroupAnswer;

	private Date createdDate;

	private String createdBy;

	private Date lastUpdatedDate;

	private String lastUpdatedBy;

	//Getters and setters
	//ToString Method

}
```

Sequence Diagram:



❖ Creating Question Option Overview

This module is used to create an encounter question option when only one checkbox is checked by the user.

A) Controller layer

EncQuestionOptionWebService.java

The file is present in the spring boot location package name: com.cpa.ehr.web.rest.system.

The method used within the class to insert the encounter question option is the *insertEncounterQuestionOptions(List<EncQuestionOptionDTO>)* method.

Input:

Name: encQuestionOptionDTO

Date-type: List

Output:

Name: headers

Data-type: ResponseEntity

```
@RestController
@RequestMapping("/api/rest/encQuestionOptions")
@CrossOrigin(origins = { "http://localhost:4300" })
public class EncQuestionOptionWebService {
	@Autowired
	private EHRBaseService ehrBaseService;

	@Autowired
	private EncQuestionOptionService
encQuestionOptionService;


	//create the QuestionOption
	@PostMapping("/createEncounterQuestionOptions")
	public ResponseEntity<EncQuestionOptionDTO>
insertEncounterQuestionOptions(@RequestBody
List<EncQuestionOptionDTO> encQuestionOptionDTO) {
		StaffMember loginUser =
ehrBaseService.currentUser();

		HttpHeaders headers= new HttpHeaders();
		for (EncQuestionOptionDTO
encQuestionOptionDTOSingleObj : encQuestionOptionDTO) {

FormatConverterUtils.setInitialDefaultValues(encQuestionOption
DTOSingleObj, loginUser);

encQuestionOptionService.persistEncounterQuestionOption(encQue
stionOptionDTOSingleObj);
		}
		return new ResponseEntity<>(headers,
HttpStatus.CREATED);
	}
```

B) Service layer

EncQuestionOptionService.java

This is an interface that contains only method names without their implementation.

The file is present in the spring boot location package name: com.cpa.ehr.service.system

The layer consists of the interface with the method for inserting the question option.

The *persistEncounterQuestionOption(encQuestionOptionDTOSingleObj)* method call is done at the service level.

Input:

Name: encQuestionOptionDTOSingleObj

Date-type: EncounterQuestionOptionDTO

Output:

EncouterQuestionOptionDTO

```
public interface EncQuestionOptionService {

	EncQuestionOptionDTO
persistEncounterQuestionOption(EncQuestionOptionDTO
encQuestionOptionDTOSingleObj);

}
```

C) Service Implementation layer

EncQuestionOptionServiceImpl.java

The class implements the interface that was described above in the Service layer, containing the concrete implementation of all the previous methods.

The logic is implemented in this class to insert the question option.

The file is present in the spring boot location package name:

com.cpa.ehr.service.system.impl

The *persistEncounterQuestionOption(EncQuestionOptionDTO encQuestionOptionDTO)* method is used to insert the question option.

Input:

Name: encQuestionOptionDTO

Date-type: EncQuestionOptionDTO

Output:

EncounterQuestionOptionDTO

```
@Service
public class EncQuestionOptionServiceImpl implements
EncQuestionOptionService {
	private static final Logger LOG =
LoggerFactory.getLogger(EncQuestionOptionServiceImpl.class);
	@Autowired
	private EncQuestionOptionMapper encQuestionOptionMapper;
	@Autowired
	private EncQuestionOptionRepository
encQuestionOptionRepository;
	@Autowired
	private EmailService emailService;

	@Override
	public EncQuestionOptionDTO
persistEncounterQuestionOption(EncQuestionOptionDTO
encQuestionOptionDTO) {
		try {
			if (encQuestionOptionDTO != null) {
				EncQuestionOption
newEncounterQuestionOption = encQuestionOptionMapper

.encQuestionOptionDTOToEntity(encQuestionOptionDTO);

				//In Built Functionality save method
used...
				EncQuestionOption
createdEncounterQuestionOption = encQuestionOptionRepository

.save(newEncounterQuestionOption);
				return (createdEncounterQuestionOption !=
null)
						?
encQuestionOptionMapper.entityToEncQuestionOptionDTO(createdEn
counterQuestionOption)
```

```
                                    : null;
                }
            } catch (Exception e) {
                String username =
SecurityUtils.getCurrentUserLogin();
                String exceptionString = "Error
persistEncounterQuestionOption{} " + "{" +
encQuestionOptionDTO + "} \n"
                            + emailService.getStackTrace(e);

emailService.sendExceptionEmail(exceptionString,username);
            }
            return null;
      }
```

D) Repository layer

encQuestionOptionRepository.java

The class consists of the repository layer which has inbuilt functionality like the save, getById, and findAll layers.

The *save()* method is called in this case to insert the question option.

The file is present in the spring boot location package name:

com.cpa.ehr.backend.dao.system

No input.

Output:

createdEncounterQuestionOption

```
@Repository
@Transactional
public interface EncQuestionOptionRepository extends
JpaRepository<EncQuestionOption, Long> {
      @Modifying
     @Query(value =
EncQuestionOptionConstants.DELETE_BY_ENCID, nativeQuery =
true)
```

E) Entity Layer

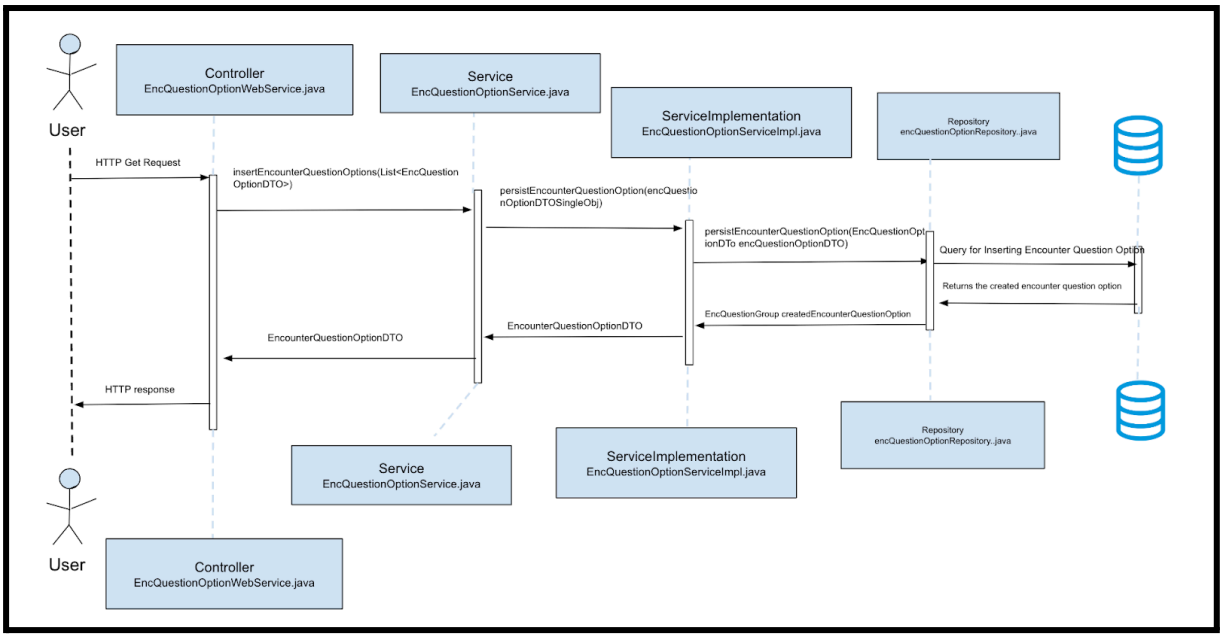
EncQuestionOptionDTO.java

The file is present in the spring boot location package name:

com.cpa.ehr.backend.service.system.dto.

The layer describes the entity attributes of the encounter question option class.

```
public class EncQuestionOptionDTO {

	private Long encQuestionOptionId;

	private Long encounterId;

	private Long questionId;

	private Long questionGroupId;

	private Long systemId;

	private Long optionId;

	private String optionValue;

	private String answer;

	private Date createdDate;

	private String createdBy;

	private Date lastUpdatedDate;

	private String lastUpdatedBy;

	private String answerComments;

	//Getters and setters

	//ToString Method
}
```


User
Controller
EncQuestionOptionWebService.java
Service
EncQuestionOptionService.java
ServiceImplementation
EncQuestionOptionServiceImpl.java
Repository
encQuestionOptionRepository. java
HTTP Get Request
insertEncounterQuestionOptions(List<EncQuestion
OptionDTO>)
persistEncounterQuestionOption(encQuestio
nOptionDTOSingleObj)
persistEncounterQuestionOption(EncQuestionOpt
ionDTo encQuestionOptionDTO)
Query for Inserting Encounter Question Option
Returns the created encounter question option
EncQuestionGroup createdEncounterQuestionOption
EncounterQuestionOptionDTO
EncounterQuestionOptionDTO
HTTP response
Repository
encQuestionOptionRepository. java
Service
EncQuestionOptionService.java
ServiceImplementation
EncQuestionOptionServiceImpl.java
User
Controller
EncQuestionOptionWebService.java