



SOFTWARE SEQUENTIAL DESIGN CONTRACT

PREPARED FOR

SAD course

PREPARED BY

Mohsen Fayyaz

Zhivar Sourati

Ali KeramtiPour

1 - Contract: Initiate_new_test

Operation:	Initiate_new_test(TestTaker t)
Cross References:	Use cases: "درخواست انجام آزمایش"
Preconditions:	<ul style="list-style-type: none"> → There is a signed in user (i.e. an instance of آزمایش دهنده : t exists and is signed in) → There should be no underway test initiations for user t → User profile is complete → There should be an instance of سوابق آزمایش associated with test taker t
Postconditions:	<ul style="list-style-type: none"> → An instance of درخواست آزمایش R was created (instance creation) → R was associated with instance of سوابق آزمایش S of test taker t. (association formed) → Attributes of R were initialized. (attribute modification) → User profile information added to attributes (attribute modification) → Initiate status of test record to "initiating" (attribute modification)

2 - Contract: prepareAllAvailableTest

Operation:	prepareAllAvailableTest()
Cross References:	Use cases: "درخواست انجام آزمایش"
Preconditions:	<ul style="list-style-type: none"> → There is at least one instance of شرح آزمایش in database
Postconditions:	<ul style="list-style-type: none"> →

3 - Contract: prepareReviewedPrescriptionsTests

Operation:	prepareReviewedPrescriptionsTests(TestTaker t)
Cross References:	Use cases: "درخواست انجام آزمایش"
Preconditions:	<ul style="list-style-type: none"> → There is a prescription associated with t (i.e. Prescription p) → Status of prescription should be approved by the expert (attr checking) → Insurance id of the prescription should be approved (attr checking)
Postconditions:	<ul style="list-style-type: none"> → →

4 - Contract: selectTests

Operation:	selectTests(List<TestDesc> TL, Bool insurance_option, Prescription p = null)
Cross References:	Use cases: "درخواست انجام آزمایش"
Preconditions:	<ul style="list-style-type: none"> → There is a Test underway. (i.e, an instance of TestRecordReq TRR exists) → There is a TestTaker T associated with TRR → If p!=null ^ insurance_option: There must be correct instance of Insurance i related with p → For all TD in TL, T is eligible to be subject of TD → There should be no duplicate TD in TL → Total number of TDs in TL should conform to tests limit rule → There should be no conflict in TDs of TL → Insurance_option must be true or false → If insurance_option is true p should not be null
Postconditions:	<ul style="list-style-type: none"> → For all TD in TL: An instance of رکورد آزمایش TR was created. (instance creation) → status of each TR was initiated (attr init)

- For each TR, TR was associated with corresponding TD in TL (Association form)
- For each TR, TR was associated with TRR (Association form)
- If P!=null: P was associated with TRR
- If P!=null: P was associated with i
- If P!=null: i was associated with TRR
- Request to use insurance attribute was modified in TRR (attr modification)

5 - Contract: retrieveUserAddresses

Operation: retrieveUserAddresses(TestTaker T)

Cross References: Use cases: "درخواست انجام آزمایش"

Preconditions: → There should be at least one instance of Address associated with T

Postconditions:

6 - Contract: setTestAddress

Operation: setTestAddress(TestRecordReq TRR, Address address)

Cross References: Use cases: "درخواست انجام آزمایش"

Preconditions:

- TRR exists and is underway.
- There must be a correct address passed in.
- Address should conform to max distance policies
- All address fields should be filled

Postconditions:

- address was associated with TRR (association form)
- testTaker profile address attr modified (attr modification)

7 - Contract: retrieveLabsWithFullSupportOverTestTakerTests

Operation: retrieveLabsWithFullSupportOverTestTakerTests(tests: TestDesc)

Cross References: Use cases: "درخواست انجام آزمایش"

Preconditions: → There should be at least one instance of tests

Postconditions:

8 - Contract: retrieveInsuranceCompaniesAssociatedWithEachLab

Operation: retrieveInsuranceCompaniesAssociatedWithEachLab(labs: List<Lab>)

Cross References: Use cases: "درخواست انجام آزمایش"

Preconditions:

Postconditions:

9 - Contract: prepareFullLabInfo

Operation: prepareFullLabInfo(labs: List<Lab>, insurances: List<Insurance>)

Cross References: Use cases: "درخواست انجام آزمایش"

Preconditions:

Postconditions:

10 - Contract: selectLab

Operation:	selectLab(TRR: TestRequestRecord, lab: Lab)
Cross References:	Use cases: "درخواست انجام آزمایش"
Preconditions:	<ul style="list-style-type: none"> → There is an underway TestRequestRecord TRR → There is a TestTaker T associated with TRR → There are instances of TestRecord TR associated with TRR → There are instances of TestDesc TD associated with each TR → There are at least one Test T associated with lab → Associated tests with lab cover all TDs → Lab distance to TRR.address should conform to distance limit rules
Postconditions:	<ul style="list-style-type: none"> → lab was associated with TRR (association form) → Corresponding lab attribute was modified in TRR (attr modification) → Credit to be paid was modified in TRR (attr modification) → TRR.status became "selected lab" → evaluatedTime was modified in TRR (attr modification)

11 - Contract: getAdvicesForUserTests

Operation:	getAdvicesForUserTests(testTaker: TestTaker, tests: List<TestDesc>)
Cross References:	Use cases: "درخواست انجام آزمایش"
Preconditions:	<ul style="list-style-type: none"> → There is an underway TestRequestRecord TRR → testTaker is associated with TRR → There should be no duplicate TD in tests
Postconditions:	

12 - Contract: approveLabAndTest

Operation: approveLabAndTest(TRR: TestRequestRecord)

Cross References: Use cases: "درخواست انجام آزمایش"

Preconditions:

- There is an underway TestRequestRecord TRR
- TRR is associated with a valid lab
- TRR is associated with a valid address

Postconditions: → TRR.status became "approved"

13 - Contract: selectTestTime

Operation: selectTestTime(TRR: TestRequestRecord, time: Time)

Cross References: Use cases: "درخواست انجام آزمایش"

Preconditions: → There is an underway TestRequestRecord TRR

Postconditions: → TRR appointment time attribute was modified (attr modification)

14 - Contract: assignPhlebotomistToTest

Operation:	assignPhlebotomistToTest(TRR: TestRequestRecord)
Cross References:	Use cases: "درخواست انجام آزمایش"
Preconditions:	<ul style="list-style-type: none"> → There is an underway TestRequestRecord TRR → There is at least 1 instances of Phlebotomist P → There is a Schedule S associated with P → S has free time in TRR.appointmentTime P has a schedule → P.location distance to TRR.address distance should conform to distance limit rule → There are at least one TestRecord testRecord associated with TRR
Postconditions:	<ul style="list-style-type: none"> → P info was modified in TRR (attr modification) → Time of appointment attribute was modified in S (attr modification) → For each test in TRR tests: an instance of testReport was created (instance creation) → For each testReport: testReport was associated with corresponding testRecord (association form) → For each testReport: testReport was associated with corresponding p(association form) → RTT.status became "p assigned"

15 - Contract: calcTotalMoneyToBePaidConsideringInsurance

Operation:	calcTotalMoneyToBePaidConsideringInsurance (TRR: TestRequestRecord)
Cross References:	Use cases: "درخواست انجام آزمایش"
Preconditions:	<ul style="list-style-type: none"> → There is an underway TestRequestRecord TRR → RTT.status is "p assigned" → There is an instance of Insurance i associated with TRR → There are at least one instances of TestRecord TR associated with TRR → There is an instance of Lab L associated with TRR → There are instances of Test T associated with L which match with TestRecords in TRR
Postconditions:	<ul style="list-style-type: none"> → The amount of money to be paid considering insurance rate to be reduced was calculated and its attribute in TRR was modified (attr modification) → TRR.paymentStatus became "ready to be paid"

16 - Contract: approveTotal

Operation:	approveTotal(TRR: TestRequestRecord)
Cross References:	Use cases: "درخواست انجام آزمایش"
Preconditions:	<ul style="list-style-type: none"> → There is an underway TestRequestRecord TRR → There is an instance of Insurance i associated with TRR → There are at least one instances of TestRecord TR associated with TRR → There is an instance of Lab L associated with TRR → TRR.paymentStatus is "ready to be paid"
Postconditions:	<ul style="list-style-type: none"> → TRR.paymentStatus became "approved to be paid" → TRR.transactionId became a unique id known by bank

17 - Contract: approvePayment

Operation:	approvePayment(transactionId: TransactionId)
Cross References:	Use cases: "درخواست انجام آزمایش"
Preconditions:	<ul style="list-style-type: none"> → There is an underway TestRequestRecord TRR → TRR.transactionId is equal to TRR transactionId → TRR.paymentStatus is "approved to be paid"
Postconditions:	<ul style="list-style-type: none"> → TRR.paymentStatus became "payment approved" → TRR.status became "payment approved"

18 - Contract: sendUserRequestDataToPhlebotomist

Operation:	sendUserRequestDataToPhlebotomist(TRR: TestRequestRecord,p: Phlebotomist)
Cross References:	Use cases: "درخواست انجام آزمایش"
Preconditions:	<ul style="list-style-type: none"> → There is an underway TestRequestRecord TRR → TRR.status is "payment approved" → All the data about TRR must be approved → testTaker address must be approved
Postconditions:	<ul style="list-style-type: none"> → TRR.status became "waiting for appointment with p" → P.freeTimes and also Schedule was updated (attr modification)

19 - Contract: userRequestInformationToLabForKitPreparation

Operation:	userRequestInformationToLabForKitPreparation(tests: List<testDesc>)
Cross References:	Use cases: "درخواست انجام آزمایش"
Preconditions:	<ul style="list-style-type: none"> → There is an underway TestRequestRecord TRR → TRR.status is "payment approved"
Postconditions:	<ul style="list-style-type: none"> → TRR.status became "notified lab to prepare the kits"