Selection Process

```
@startuml
[*] --> Created



state "Created" as Created
note right of Created: status = notViewed/Matchmaking; tag = Sent/Received

Created --> Viewed : "CompanyTutor/Student views => status=Viewed"



state "Viewed" as Viewed
note right of Viewed: Could still be refused by either side; Or accepted => next steps

Viewed --> Refused : "Student or Company Tutor refuses"
Viewed --> Accepted : "Both accept"



state "Refused" as Refused
note right of Refused: Process ends

Refused --> [*]

state "Accepted" as Accepted
note right of Accepted: Accept => Schedule/set meeting

Accepted --> Meeting: "Meeting scheduled => Meeting_set"



state "Meeting" as Meeting
note right of Meeting: MeetingSet

Meeting --> Questionnaire : "Meeting done => QuestionnaireCompanyTutor"



state "FirstQuestionnaire" as Questionnaire
note right of Questionnaire: Tutor fills and upload the MeetingQuestionnaire

Questionnaire --> DecisionStudent : "Complete Questionnaire => Decision_student"
```

```
state "DecisionStudent" as DecisionStudent
note right of DecisionStudent: Student can Refuse, Call_again, or Accept

DecisionStudent --> Refused : "Student_reject"
DecisionStudent --> Meeting : "Student_call_again => set new meeting"
DecisionStudent --> AcademicTutor : "Student_accept => choose tutor"



state "AcademicTutor" as AcademicTutor
note right of AcademicTutor: Chosen tutor can accept or refuse

AcademicTutor --> DecisionStudent : "Tutor refuses offer => Student has to pick another one"
AcademicTutor --> FinalDecisionCompany: "Tutor accept => FinalDecisionCompany"



state "FinalDecisionCompany" as FinalDecisionCompany
note right of FinalDecisionCompany: - If refuse => ends, \n- If accept => stage is activated

FinalDecisionCompany --> ProcessConcluded : "Accept => active stage"
FinalDecisionCompany --> Refused : "Refuse => ends"



state "ProcessConcluded" as ProcessConcluded
note right of ProcessConcluded: 1. All other selectionProcesses for this offer => refused; \n2.
Stage & user moves to an ActiveStage

ProcessConcluded --> [*]

@enduml
```

Stage Lifecycle

```
@startuml
[*] --> OfferCreation

state OfferCreation as "OfferCreation"
note right of OfferCreation: The tutor can: \n- choose a template \n- edit/fill the form \n- improve the content

OfferCreation --> Draft : saveAsDraft
OfferCreation --> Published : publishOffer
OfferCreation --> Deleted : deleteOffer



state Draft as "Draft"
note right of Draft: The tutor can: \n- access the draft \n- publish it \n- delete it

Draft --> OfferCreation: modifyDraft
Draft --> Published : publishOffer
Draft --> Deleted : deleteOffer



state Deleted as "DELETED"
note right of Deleted: Offer is removed from the system

Deleted --> [*]



state Published as "PUBLISHED"
note right of Published: Visible for: \n- matchmaking \n- selection process

Published --> OfferCreation: modifyDraft
Published --> Deleted : deleteOffer
Published --> [*]: stageCompleted

@enduml
```

Active Stages

```
@startuml
[*] --> ActiveStageCreated : "Stage Created"

state "Active Stage Created" as ActiveStageCreated
note right of ActiveStageCreated: - status = IN_PROGRESS \n- startDate set

ActiveStageCreated --> InProgress : "Normal progression of the stage"



state "In Progress" as InProgress
note right of InProgress: Can: \n- schedule events (CalendarEvent) \n- encounter problems (Problem Chat)

InProgress --> Suspended : "Problem arises => Suspend"
InProgress --> FinalEvaluation : "LastEvent flag triggered => Final Evaluation"
InProgress -left-> InProgress: "New events"



state "Suspended" as Suspended
note right of Suspended: Temporarily halted by tutor

Suspended --> InProgress : "Resume"
Suspended --> Terminated : "Severe problem"



state "Terminated" as Terminated
note right of Terminated: - Stage is ended \n- endDate is set \n- status = TERMINATED

Terminated --> [*]



state "Final Evaluation" as FinalEvaluation
note right of FinalEvaluation: Final evaluation from all parties are filled

FinalEvaluation --> Completed : "All final evaluations done => Exchanged => Completed"



state "Completed" as Completed
```

```
note right of Completed: - Stage is closed \n- endDate is set \n- status = COMPLETED

Completed --> [*]

@enduml
```

Domain association

```
@startuml
[*] --> Start

state Start
note right of Start : No User

Start --> RegistrationInProgress : register

state RegistrationInProgress
note right of RegistrationInProgress : User fills form, uploads CV, etc.

RegistrationInProgress --> CheckDomain : submit registration

state CheckDomain
note right of CheckDomain : Check Domain:\n- Known & Verified: Associate user to Institution;\n- Known but NOT Verified: User -> Pending;\n- New Domain: Must be verified by admin/tutor;

CheckDomain --> DomainVerified : Domain Known & Verified
CheckDomain --> PendingVerification : Domain Known but NOT Verified / New Domain

state DomainVerified
note right of DomainVerified : User Active & Verified:\nuser.isActive = true

state PendingVerification
note right of PendingVerification : Waiting for Domain Verification:\nuser.isVerified = false;\nuser sees 'Verification in Progress'

PendingVerification --> DomainVerified : Domain Verified\nuser.isVerified = true\nuser.isActive = true (if domain ok)

DomainVerified --> [*] : User Active & Verified

@enduml
```