**Student::createProfile()**

* Prompt the user for name and email
* Initialize the student object with the provided data
* Optionally ask the user to add initial courses
* Add selected Course objects to the courses vector

**Student::updateProfile()**

* Allow editing of the student's name and email
* Optionally allow editing of course list (add/remove)

**Student::addAvailability(slot)**

* Add the given Availability object to the student’s availability vector

**Student::removeAvailability(slot)**

* Search for and remove the matching Availability slot from the vector

**Student::searchBuddies(course)**

* For each other student:
  + If they are enrolled in the same course
  + And if any of their availability slots match the current student's
    - Then add them to the result list
* Return the list of matching Student\*

**Student::receiveMatchSuggestions()**

* Loop through all courses in student's course list
* For each course:
  + Use searchBuddies(course) to find potential matches
* Aggregate and return all unique matches

**Student::scheduleSession(session)**

* Add the session pointer to the student’s scheduledSessions vector
* Call sendInvitation() on the Session object

**Student::confirmSession(session)**

* Call confirm() on the Session object

**Student::declineSession(session)**

* Call decline() on the Session object

**Session::sendInvitation()**

* Set status to Pending
* Notify all participants that an invitation was sent

**Session::confirm()**

* Set status to Confirmed
* Notify both participants of the confirmation

**Session::decline()**

* Set status to Declined
* Notify both participants of the decline