

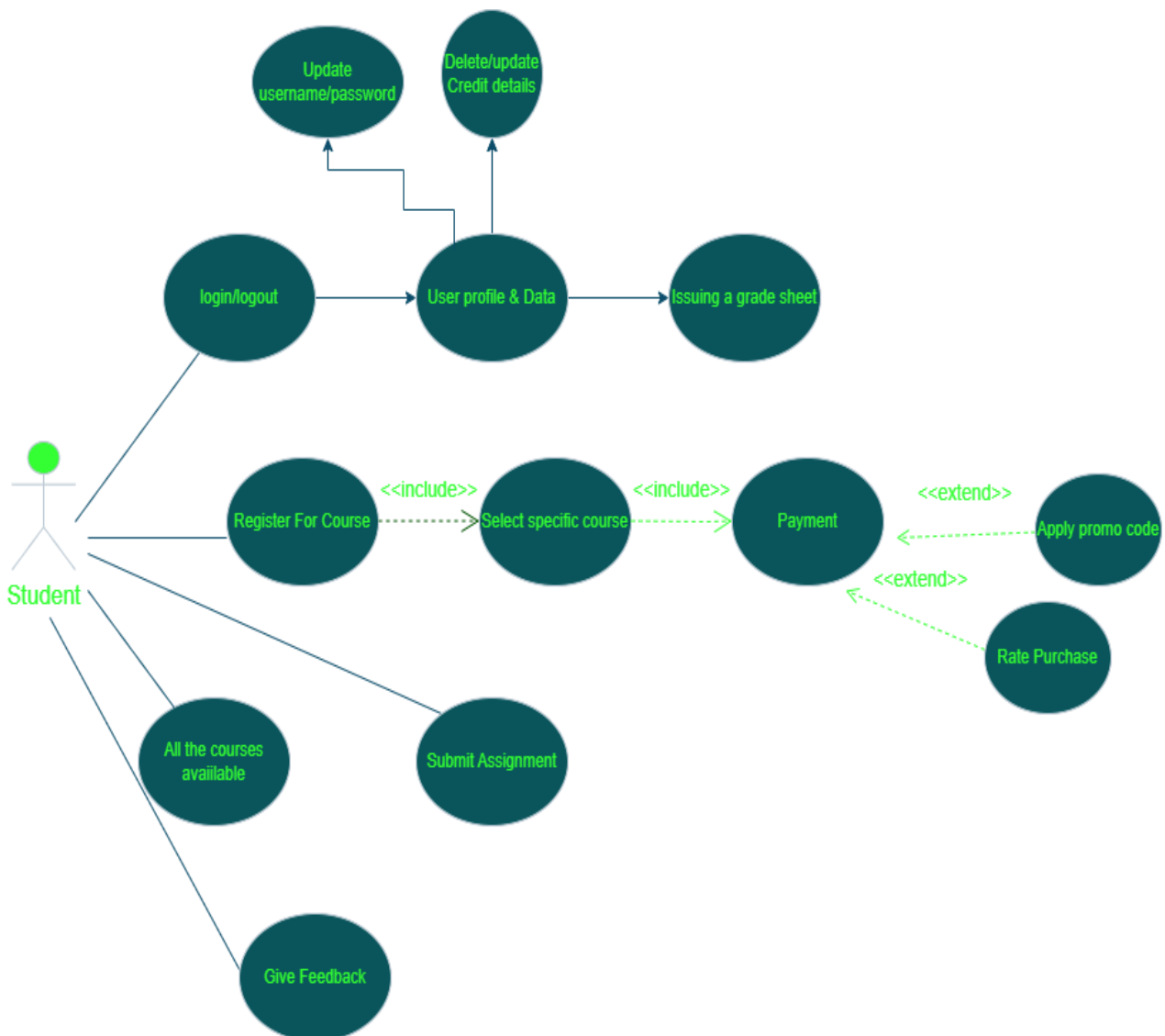
USE-CASE DIAGRAM 1

ADMIN

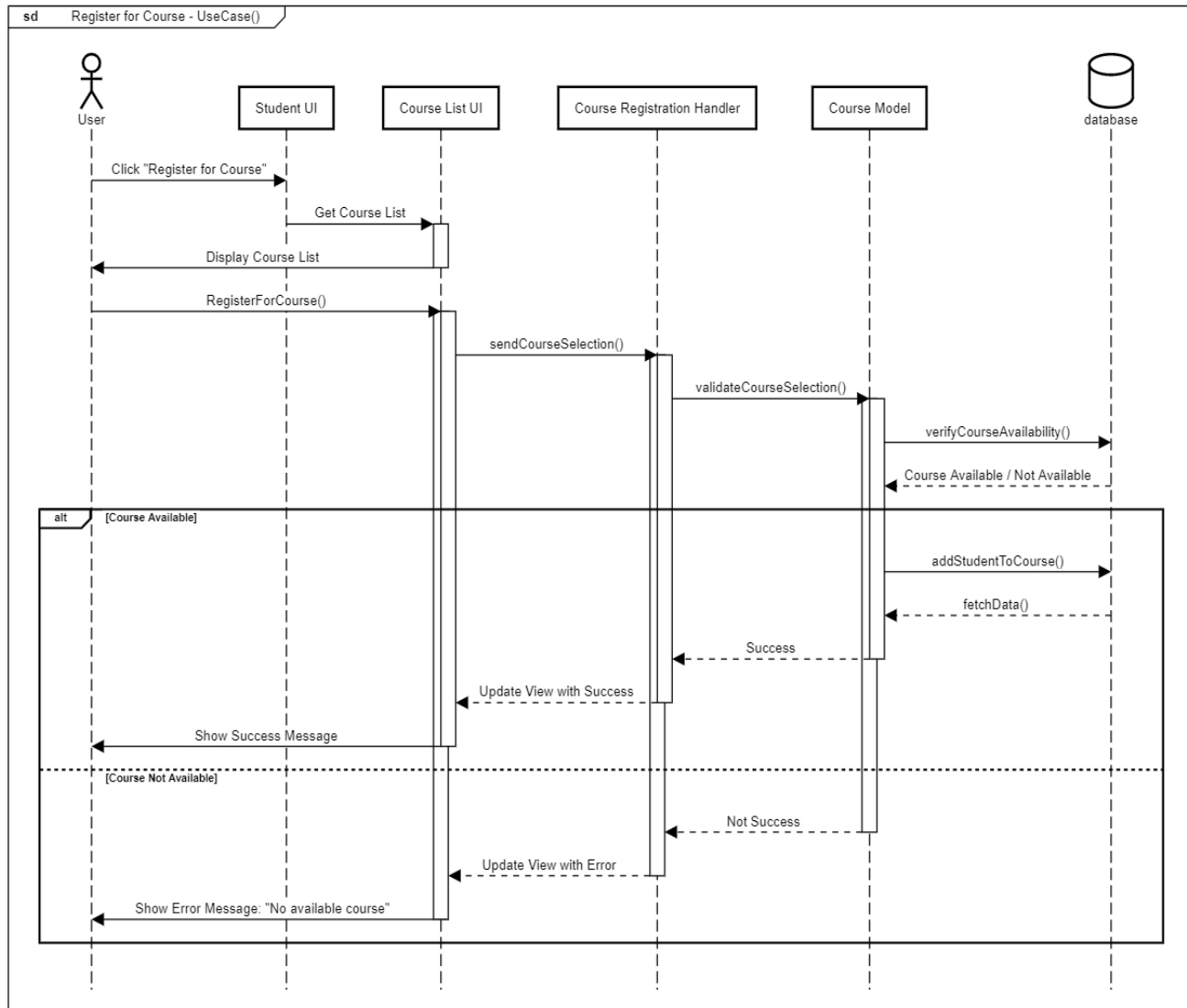


USE-CASE DIAGRAM 2

STUDENT



SEQUENCE DIAGRAM 1



frame Register for Course – UseCase()

actor User

User -> Student UI: Click "Register for Course"

Student UI -> Course List UI: Get Course List

activate Course List UI

Course List UI -> User: Display Course List

deactivate Course List UI

User -> Course List UI: RegisterForCourse()
activate Course List UI
activate Course List UI
Course List UI -> Course Registration Handler: sendCourseSelection()
activate Course Registration Handler
activate Course Registration Handler
Course Registration Handler -> Course Model: validateCourseSelection()
activate Course Model
activate Course Model
database database
Course Model -> database: verifyCourseAvailability()
database --> Course Model: Course Available / Not Available
alt Course Available
Course Model -> database: addStudentToCourse()
database --> Course Model: Confirmation
Course Model --> Course Registration Handler: Success
deactivate Course Model

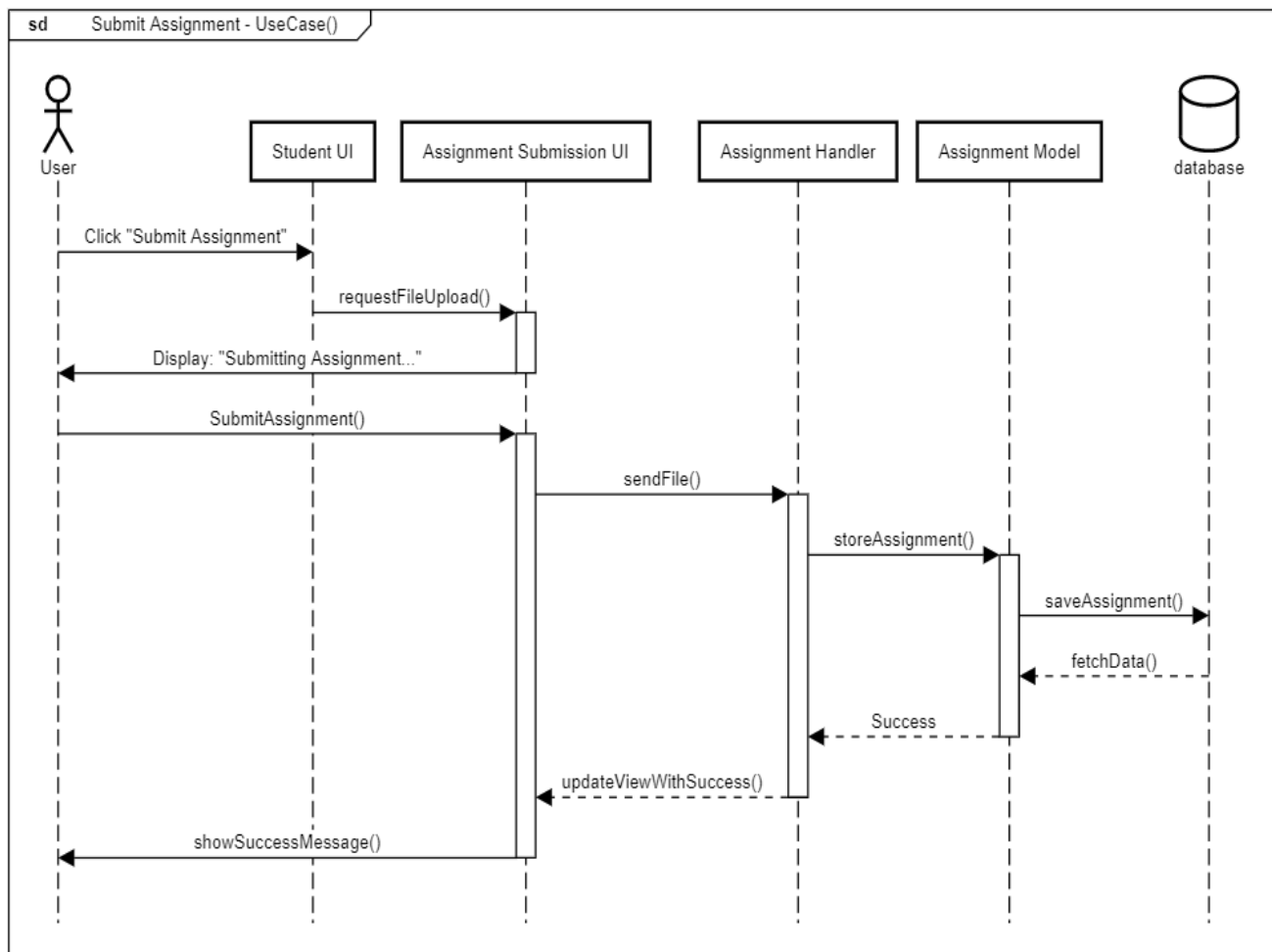
Course Registration Handler --> Course List UI: Update View with Success
deactivate Course Registration Handler

Course List UI -> User: Show Success Message
deactivate Course List UI
else Course Not Available
Course Model --> Course Registration Handler: Not Success
deactivate Course Model

Course Registration Handler --> Course List UI: Update View with Error
deactivate Course Registration Handler
Course List UI -> User: Show Error Message: "No available course"
deactivate Course List UI

end

SEQUENCE DIAGRAM 2



frame Submit Assignment – UseCase()

actor User

User -> Student UI: Click "Submit Assignment"

Student UI -> Assignment Submission UI: requestFileUpload()

activate Assignment Submission UI

Assignment Submission UI -> User: Display: "Submitting Assignment..."

deactivate Assignment Submission UI

User -> Assignment Submission UI: SubmitAssignment()

activate Assignment Submission UI

Assignment Submission UI -> Assignment Handler: sendFile()

activate Assignment Handler

Assignment Handler -> Assignment Model: storeAssignment()

activate Assignment Model

database database

Assignment Model -> database: saveAssignment()

database --> Assignment Model: Success

Assignment Model --> Assignment Handler: Success

deactivate Assignment Model

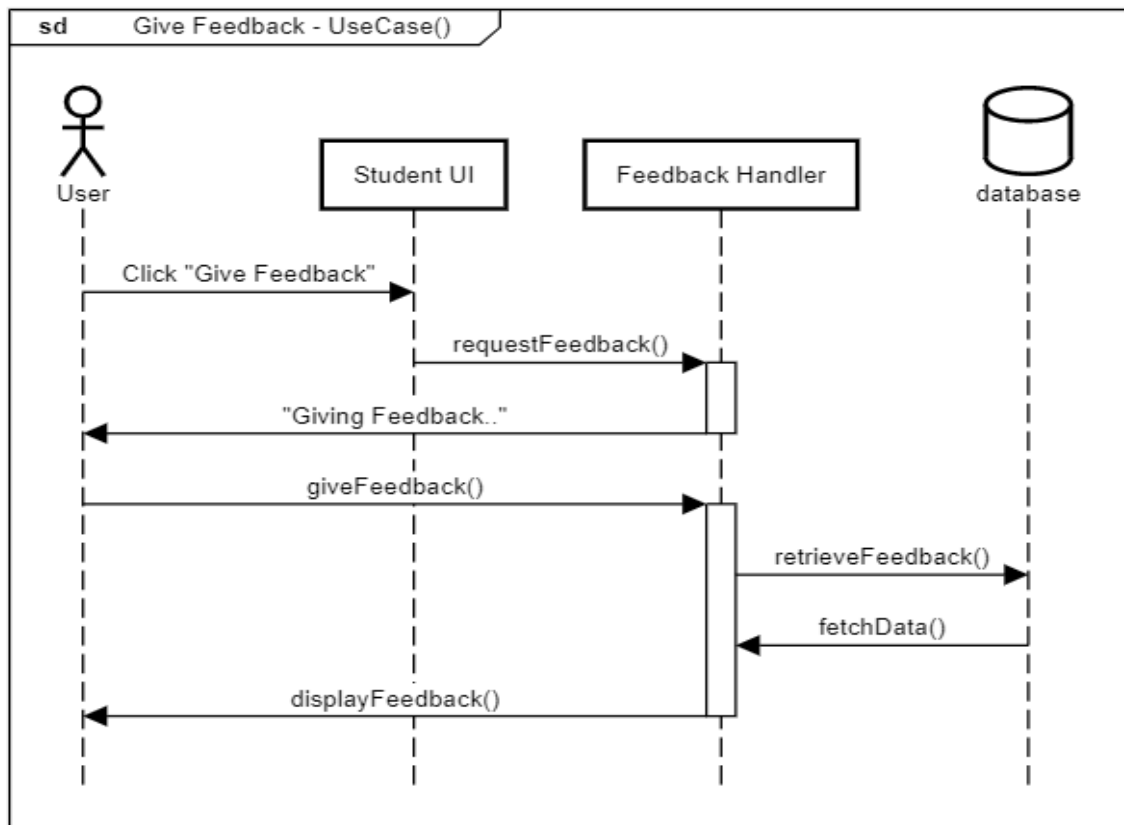
Assignment Handler --> Assignment Submission UI: updateViewWithSuccess()

deactivate Assignment Handler

Assignment Submission UI -> User: showSuccessMessage()

deactivate Assignment Submission UI

SEQUENCE DIAGRAM 3



frame Give Feedback – UseCase()

actor User

User -> Student UI: Click "Give Feedback"

Student UI -> Feedback Handler: requestFeedback()

activate Feedback Handler

Feedback Handler -> User: "Giving Feedback..."

deactivate Feedback Handler

User -> Feedback Handler: giveFeedback()

activate Feedback Handler

database database

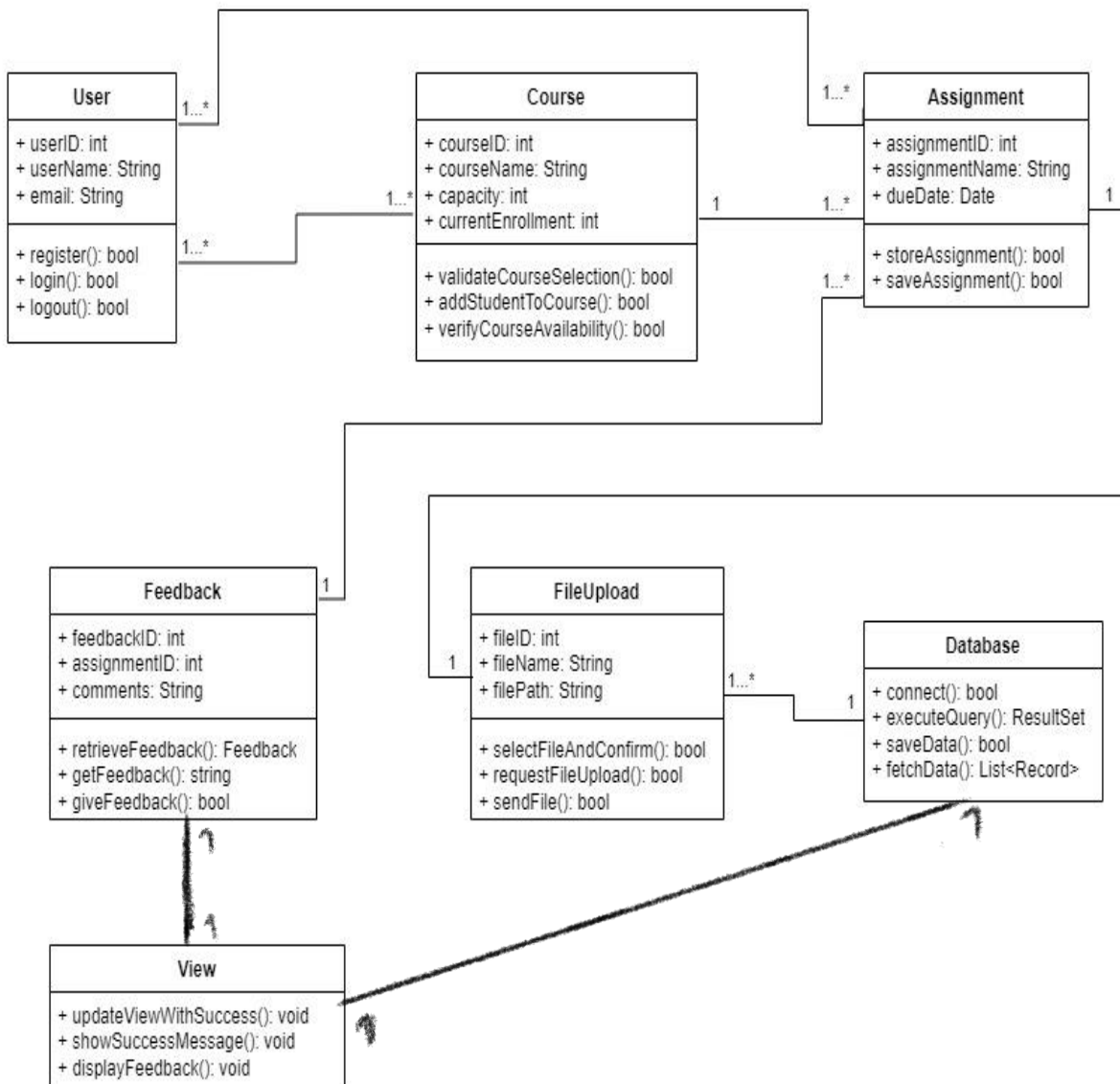
Feedback Handler -> database: retrieveFeedback()

database -> Feedback Handler: fetchData()

Feedback Handler -> User: displayFeedback()

deactivate Feedback Handler

Class Diagram for TeachMate System



MVC MODEL

Model:

דוגמאות למחלקות ולפונקציות בשכבת המודל

- User: (register(), login(), logout()). כניסה ויציאה.
- Course: (validateCourseSelection(), addStudentToCourse(), registerForCourse()). ניהול הקורסים כולל בדיקת זמינות, הוספת סטודנט לקורס ורישום לקורס.
- Assignment: (storeAssignment(), saveAssignment(), submitAssignment()). ניהול המטלות והגשה.
- Feedback: (retrieveFeedback(), requestFeedback()). ניהול המשוב על מטלות והגשת בקשות למשוב.

View:

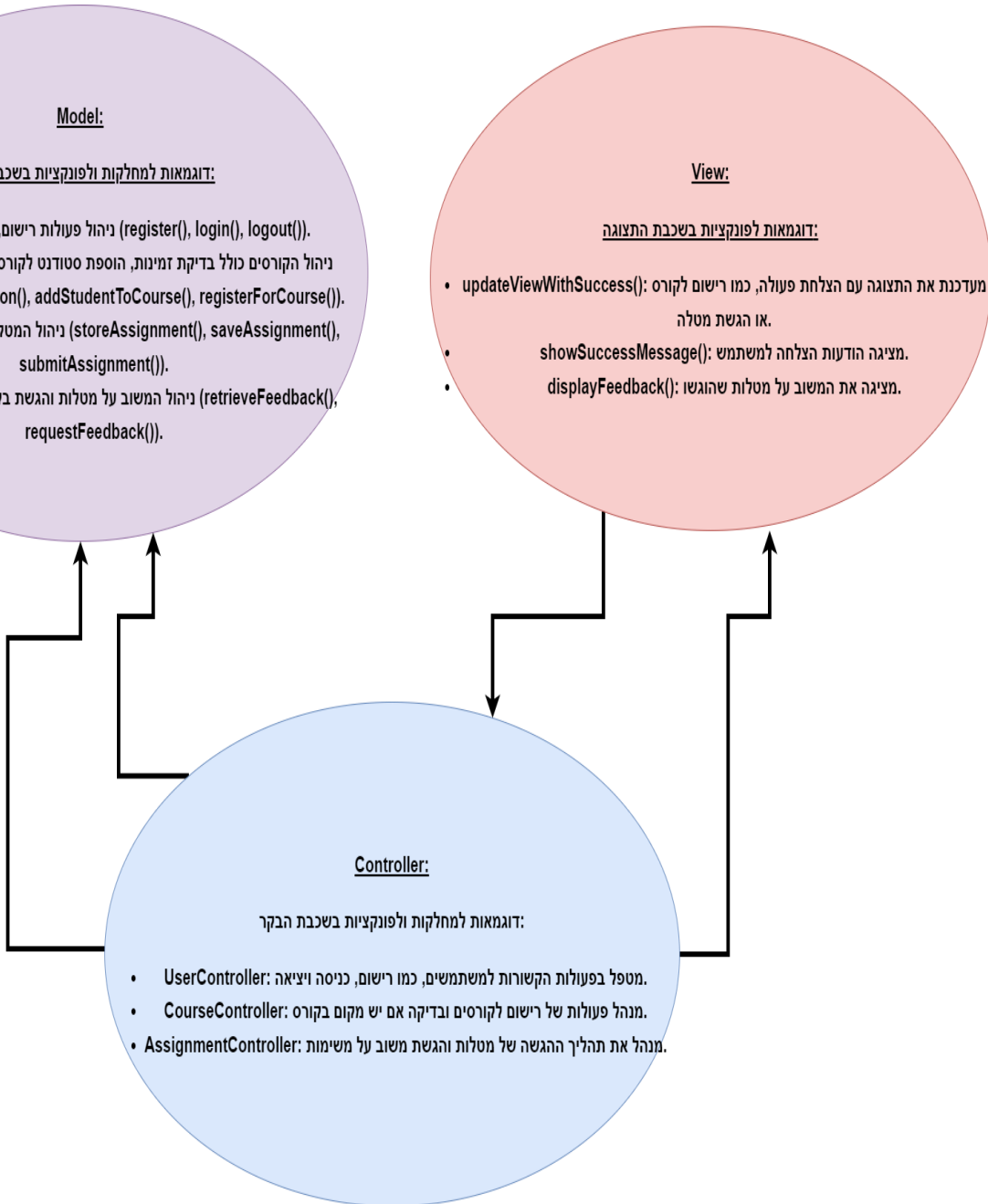
דוגמאות לפונקציות בשכבת התצוגה

- updateViewWithSuccess(): מעדכנת את התצוגה עם הצלחת פעולה, כמו רישום לקורס. מציגה הודעות הצלחה למשתמש.
- showSuccessMessage(): מציגה את המשוב על מטלות שהוגשו.
- displayFeedback(): או הגשת מטלה.

Controller:

דוגמאות למחלקות ולפונקציות בשכבת הבקר

- UserController: מטפל בפעולות הקשורות למשתמשים, כמו רישום, כניסה ויציאה.
- CourseController: מנהל פעולות של רישום לקורסים ובדיקה אם יש מקום בקורס.
- AssignmentController: מנהל את תהליך ההגשה של מטלות והגשת משוב על משימות.



Model:

המודל אחראי על ניהול הנתונים המרכזיים של המערכת, כולל משתמשים, קורסים, מטלות ומשוב. המודל מתממשק עם בסיס הנתונים ומאפשר שליפת נתונים ועדכונם.

כיצד זה עונה על צרכי האפיון של המערכת:

המודל עונה על צרכי האפיון בכך שהוא אחראי על ניהול הנתונים המרכזיים במערכת כמו פרטי המשתמשים, הקורסים והמשימות. הוא מתממשק עם בסיס הנתונים ומוודא שהנתונים נכונים ומאורגנים בצורה שמאפשרת לעקוב אחריהם ולעדכן אותם, כך שהמערכת תמיד תציג את המידע העדכני.

View:

שכבת התצוגה אחראית על הצגת המידע בצורה ידידותית ומותאמת למשתמש. התצוגה מקבלת את הנתונים מהמודל ומציגה אותם למשתמש בצורה נוחה וברורה.

כיצד זה עונה על צרכי האפיון של המערכת:

שכבת התצוגה עונה על הצרכים של המערכת שלך בכך שהיא מציגה למשתמשים את המידע בצורה ידידותית וברורה. הסטודנטים יכולים לצפות בקורסים, בשיעורים ובמשימות שלהם בקלות, והמורים יכולים לנהל את הקורסים בצורה יעילה. זה משפר את חוויית המשתמש, מה שהוגדר כאחת הדרישות המרכזיות במערכת.

Controller:

הבקר אחראי על קישור בין המשתמשים לבין המודל והתצוגה. הוא מקבל את הקלט מהמשתמשים, מעבד אותו ומפעיל את הלוגיקה הדרושה כדי לעדכן את המודל ולהציג את הנתונים בתצוגה.

כיצד זה עונה על צרכי האפיון של המערכת:

הבקר עונה על צרכי האפיון בכך שהוא אחראי על קבלת הקלט מהמשתמשים, כמו למשל הגשת משימות או רישום לקורסים, ומעביר את המידע הזה למודל לצורך עדכון בסיס הנתונים. הבקר גם מבצע את הלוגיקה הדרושה כדי לעדכן את התצוגה ולהציג את המידע המעודכן למשתמשים בזמן אמת.