

# פרויקט סיום הנדסת תוכנה

מוגש ל: יניב לוי

מספר קורס: 63301

קבוצת תרגול מס' 62-663301



# מערכת אונליין לשיעורים פרטיים

## תיאור הפרויקט:



"Easy Learn" היא פלטפורמה אינטרנטית ייעודית לשיעורים פרטיים וקבוצתיים. המערכת מיועדת לסטודנטים ממוסדות להשכלה גבוהה. מטרת הפלטפורמה היא לסייע לסטודנטים להעמיק ולהעשיר את הידע שלהם בתחומים שונים. הפלטפורמה מעניקה גישה נוחה מכל מקום, כך שהסטודנטים יכולים להשתתף בשיעורים לפי נוחיותם בזמן המתאים ביותר עבורם, גם מחוץ לשעות הלימודים הרשמיות. בנוסף, "Easy Learn" מציעה ממשק ידידותי המותאם לצרכים של הסטודנטים והמרצים כאחד ומספקת סביבת למידה עוטפת, תומכת ונגישה לכל סטודנט.

## חקר ישימות:



המערכת "Easy Learn" מותאמת לשילוב במוסדות להשכלה גבוהה ומספקת מענה לסטודנטים הזקוקים לתמיכה נוספת מעבר לשיעורים הרגילים. המערכת מאפשרת השתתפות בשיעורים פרטיים וקבוצתיים בהתאמה אישית לצרכים הלימודיים, בכל זמן ומקום. היא תומכת בניהול מערך הלימודים בשגרה ובחירום, תוך אינטגרציה עם מערכות ניהול הלמידה הקיימות וסנכרון עם לוחות הזמנים של הסטודנטים והמרצים. ההתחברות נעשית דרך משאבי מוסד הלימוד, מה שמרחיב את מגוון הכלים הזמינים. בנוסף, היא מתממשת ל-ZOOM ולשרתים המכילים הקלטות וחומרי לימוד לשימוש עתידי.

## בעלי העניין:



### 1. סטודנטים : (בעלי עניין ישירים)

הם המשתמשים העיקריים של המערכת.

**השפעה חיובית:** לסטודנטים תהיה פלטפורמה נוחה ומקוונת איתה יוכלו ללמוד מרחוק ומכל מקום. הסטודנטים מחפשים פתרונות יעילים לשיפור הידע, הבנת החומר והכנה למבחנים, המערכת נועדה לספק להם את הפתרונות הללו ובנוסף מספקת גישה נוחה שמותאמת לצרכיהם האישיים.

**השפעה שלילית:** חוסר שימוש או חוסר שביעות רצון מצד הסטודנטים יכול להוביל לכישלון המערכת.

### 2. מרצים / מתרגלים : (בעלי עניין ישירים)

**השפעה חיובית:** המערכת מעניקה למרצים את היכולת לתזמן את שעות הקבלה והתגבורים לפי העדפותיהם האישיות, ולפקח על מספר הסטודנטים שנרשמו לכל מפגש, כך שיוכלו להתכונן בהתאם.

**השפעה שלילית:** מרצים/מתרגלים שלא מסתגלים לטכנולוגיה או שמתקשים להשתמש במערכת יכולים להוביל לבעיות בהטמעתה. התנגדות מצידם יכולה לעכב את הפרויקט.

3. **מוסדות הלימוד להשכלה גבוהה** (בעלי עניין ישירים)  
**השפעה חיובית:** המערכת תסייע לסטודנטים בשמירה על קצב החומר הנלמד, תגביר את סיכוייהם להצליח, ובכך תשפר את הרמה האקדמית של מוסד הלימוד.  
**השפעה שלילית:** סטודנטים עלולים להוריד את הקלטות השיעורים ובכך לפגוע בזכויות היוצרים של המוסד האקדמי ושל המערכת.

4. **שוק המורים הפרטיים** (בעלי עניין עקיפים)  
**השפעה שלילית:** המערכת מספקת שעות תרגול נוספות בנוסף לשיעורים המוצעים במערכת, מה שיגרום לסטודנט לא בהכרח להזדקק לשיעורים פרטיים באופן עצמאי על-ידי מורים חיצוניים, דבר שמשפיע על שוק המורים הפרטיים.

5. **חברות מתחרות** (בעלי עניין עקיפים)  
**השפעה שלילית:** בעולם טכנולוגיות הלמידה קיימים פתרונות מולטימדיה רבים, המספקים למוסדות הלימוד אפשרויות דומות. פיתוח ויישום מערכות כאלה במוסדות אלה עשויים להחריף את התחרות בשוק, ובכך לפגוע בהכנסות של החברות המתחרות.

6. **המשקיעים** (בעלי עניין עקיפים)  
**השפעה חיובית:** המשקיעים אינם מושפעים ישירות מהמערכת, אך עשויים לתמוך בפרויקטים חינוכיים ולסייע במימון. הם מעוניינים בהצלחה כלכלית ושיפור איכות החינוך, ומחפשים החזר על ההשקעה ורווחים מהמערכת.  
**השפעה שלילית:** המשאבים הכספיים והמומחיות העסקית שלהם חיוניים ולכן חוסר תמיכה עלול לעכב את הפיתוח בשל מחסור במשאבים.

### דרישות פונקציונליות:



1. גישה דרך חשבון המוסד: המערכת לא מאפשרת יצירת חשבון משתמש חדש, רק מי שהינו חלק מהמוסד הלימודי יוכל להתחבר למערכת ולהשתמש בה. הסטודנטים והמרצים יתחברו למערכת באמצעות פרטי ההתחברות שלהם למוסד הלימוד.
2. רישום לשיעורים: סטודנטים יוכלו להירשם לשיעורים פרטיים/קבוצתיים באמצעות פרטי המשתמש שלהם באמצעות הלשוניות המיועדות.
3. תיאום עם מערכת שעות: המערכת תתאם בין השיעורים הפרטיים ללוח הזמנים האקדמי של הסטודנטים והמרצים ולא תאפשר לקבוע שיעורים שחופפים לשיעורים הקבועים במערכת.
4. היסטוריית קורסים: המערכת תאפשר לסטודנטים לצפות בהיסטוריית הקורסים שלהם ולבחור שיעורים פרטיים בהתבסס על קורסים שנלמדו או נלמדים.
5. התראות למרצים ולסטודנטים: המערכת תשלח התראות על שיעורים קרובים, עדכונים ושינויים בהזמנות.
6. גישה להקלטות שיעורים: המערכת תציג לסטודנט רשימת שיעורים מוקלטים.

7. ממשק אישי מותאם: המערכת תספק ממשק אישי מותאם לכל סטודנט ומרצה לפי הצרכים שלהם והגדרות המוסד.
8. הגדרת זמינות המרצים: המרצים יוכלו להגדיר את זמינותם לשיעורים פרטיים וסטודנטים יוכלו לבחור שעות לפי הזמינות.
9. תמיכה בתקלות: המערכת תכלול אפשרות ליצירת קשר עם שירותי התמיכה של המערכת במקרה של בעיות טכניות או שאלות.
10. סוג שיעורים: קיימים במערכת שני סוגים של שיעורים:  
שיעורים קבוצתיים: הרשמה לשיעורים קבוצתיים תתאפשר תחת לשונית "שיעורים קבוצתיים פנויים". שיעורים קבוצתיים יוגבלו עד ל-10 משתתפים.  
שיעורים פרטיים: הרשמה לשיעורים פרטיים תתאפשר תחת לשונית "שיעורים פרטיים פנויים". שיעורים פרטיים יוגבלו לסטודנט אחד.
11. מנוע חיפוש: המערכת תספק תכונת חיפוש המאפשרת למשתמשים למצוא שיעורים על סמך מילות מפתח (שם מרצה/מתרגל, שם קורס, סוג שיעור- פרטי או קבוצתי, תאריך).
12. רשימת המתנה: לסטודנט המעוניין בשיעור שנתפס, המערכת תאפשר להירשם לרשימת המתנה. במידה ויתפנה מקום לשיעור, יקבל הסטודנט הממתין הודעה על מקום שהתפנה כדי שיוכל לבצע רישום דרך המערכת.
13. צ'אט: במערכת יהיה צ'אט אישי מובנה בו כל סטודנט יוכל לשוחח עם המרצה איתו קיים או יקיים את השיעורים. הסטודנט יוכל לשאול שאלות בנוגע לחומר השיעור והמרצה יוכל לשלוח בצ'אט סיכומים ולענות על שאלות הסטודנט.
14. בדיקת נוכחות: בתחילת כל שיעור למרצה 'תקפוף' חלונית שבה יצטרך לבדוק נוכחות של הסטודנטים שהתחברו לשיעור מתוך הנרשמים.
15. הגבלת כמות שיעורים: לכל קורס יוכל הסטודנט לקבוע עד 3 שיעורים בשבוע.
16. בקרת גישה והרשאות: למרצה תהיה אפשרות להגדיר גישה לסטודנטים לחומרים מסוימים או אזורים במערכת. לסטודנט תהיה אפשרות להגיש בקשה לקבלת גישה לתכנים מסוימים.

## דרישות לא פונקציונליות:



### : Security Requirement

1. בטיחות נתונים: המערכת תעבוד לפי תקני אבטחת מידע מחמירים ותגן על פרטי המשתמשים מפני גניבה או גישה לא מורשית, על ידי פרוטוקול HTTPS.

### : Reliability Requirement

2. שחזור נתונים: המערכת תספק פתרונות גיבוי ושחזור כדי להבטיח שמירה על הנתונים במקרה של תקלות טכניות או אובדן נתונים, על ידי ענן Azure של Microsoft.
3. זמינות גבוהה: המערכת תהיה זמינה 24/7 עם רמת זמינות של לפחות 99.9%.

### : Performance Requirement

4. ביצועים: המערכת תספק זמן תגובה מהיר תוך פחות מ-2 שניות.

: Scalability Requirement

5. תמיכה בעומסים: המערכת תתמוך בעומס גבוה של 10,000 משתמשים בו זמנית, במיוחד בתקופות מבחנים והגשת מטלות כאשר מספר המשתמשים עלול לגדול משמעותית.

: Portability Requirement

6. התאמה למכשירים שונים: המערכת תהיה מותאמת לשימוש בדפדפנים שונים כגון Chrome, Firefox, Edge ובמערכות הפעלה Windows ו-Mac. בנוסף, המערכת תתמוך בהתחברות ממחשבים, טאבלטים, אייפדים וטלפונים חכמים.

: Regulatory Requirement

7. נגישות: המערכת תתמוך בגישה עבור אנשים עם מוגבלויות על-פי התקן הישראלי 5568, כולל התאמות למערכות קוראי מסך ויכולת לשנות את גודל הטקסט.

: Space Requirement

8. אחסון: המערכת צריכה לספק לפחות 15 טרה-בייט של נפח אחסון לכלל המשתמשים לצורך שמירת שיעורים, מסמכים, וידאו ונתונים נוספים.

: Development Requirement

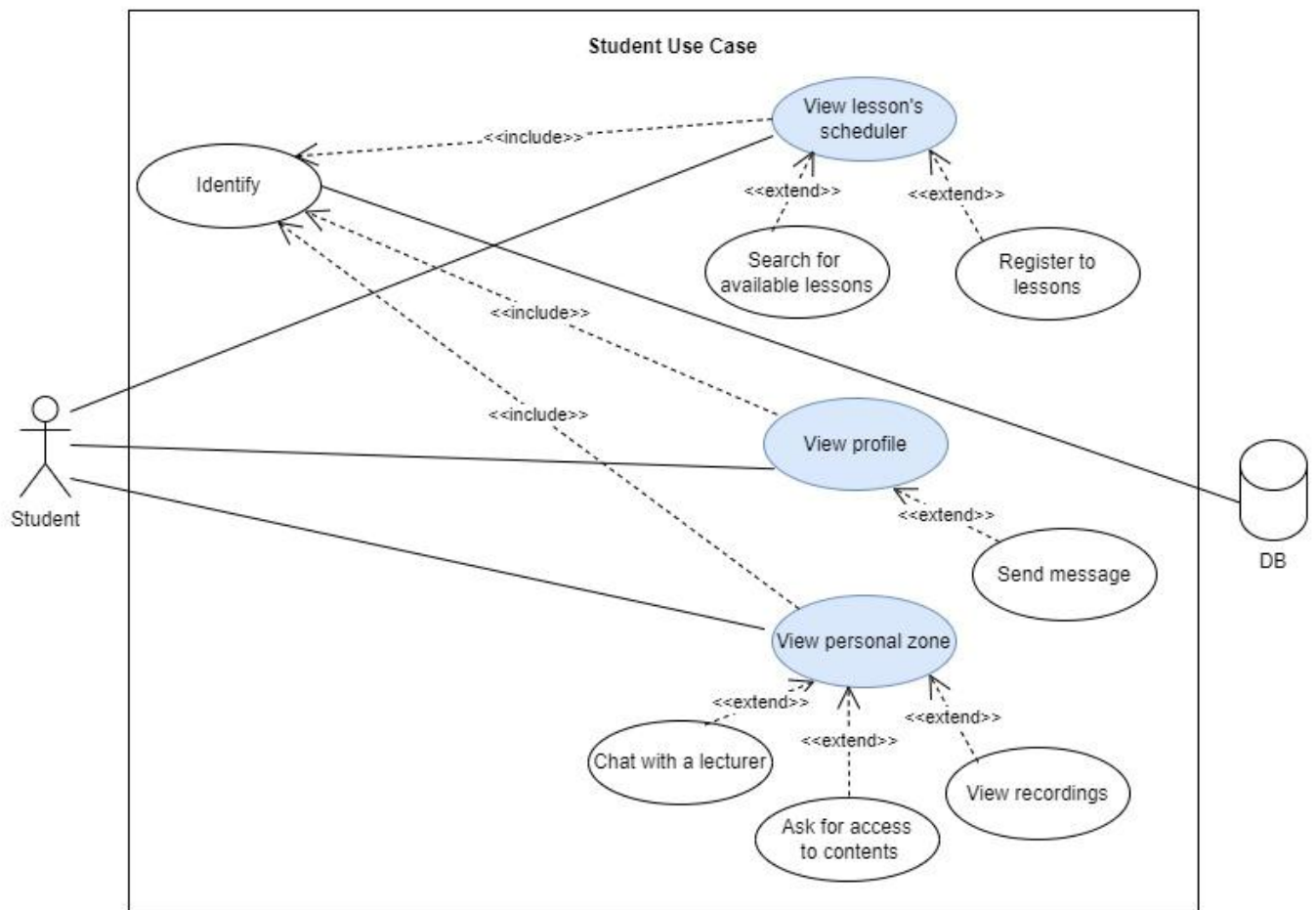
9. תיעוד: המערכת תספק תיעוד מסודר למשתמשים ולמפתחים, כולל מדריכי משתמש, מדריכים טכניים ודוקומנטציה על התהליכים.

: Serviceability Requirement

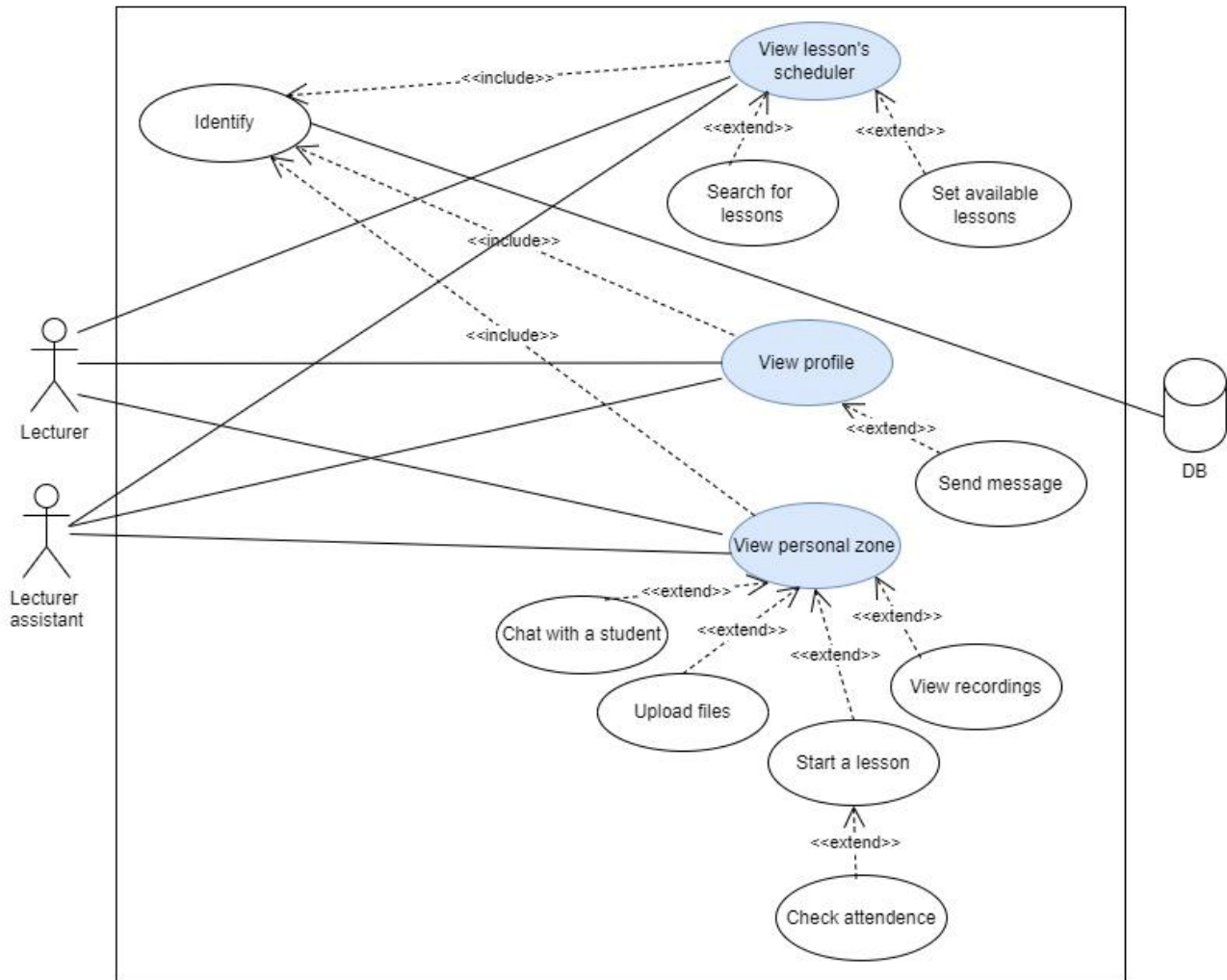
10. טיפול בכשלים: המערכת תספק מנגנון לניהול ודיווח כשלים תוך רישום כל השגיאות.

: Usability Requirement

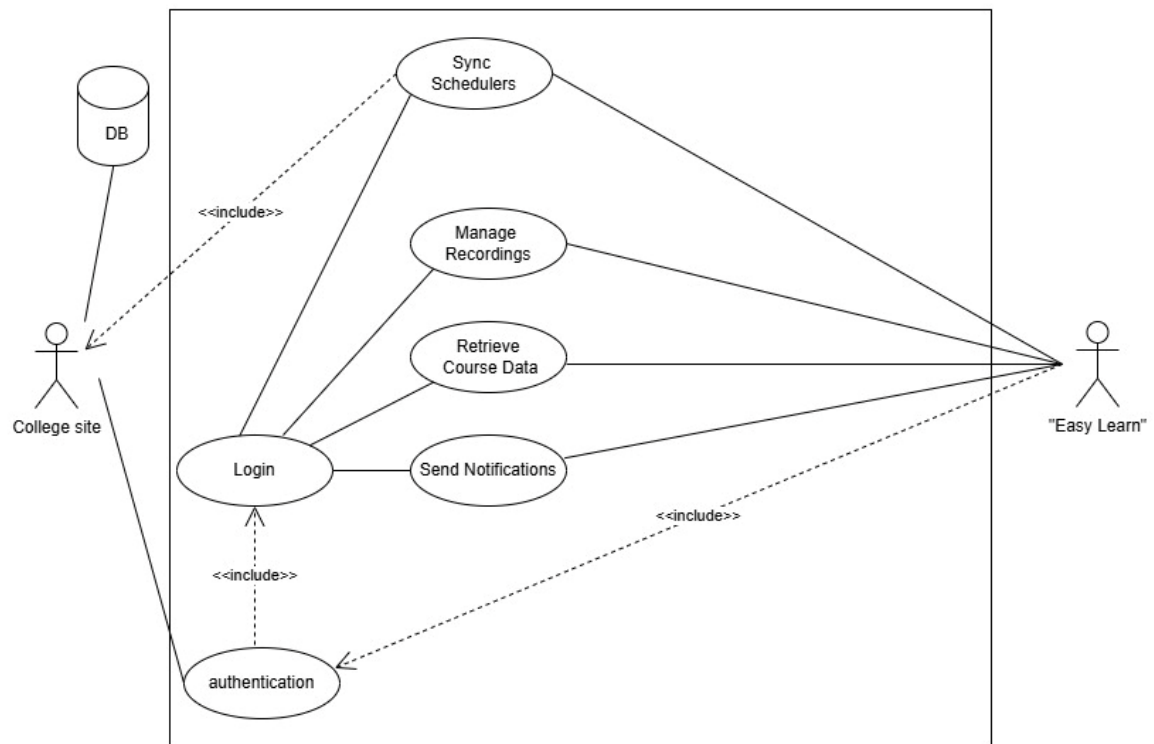
11. תמיכה בשפות: המערכת תתמוך בשפות הבאות: עברית, אנגלית, ערבית ורוסית.



# Lecturer Use Case

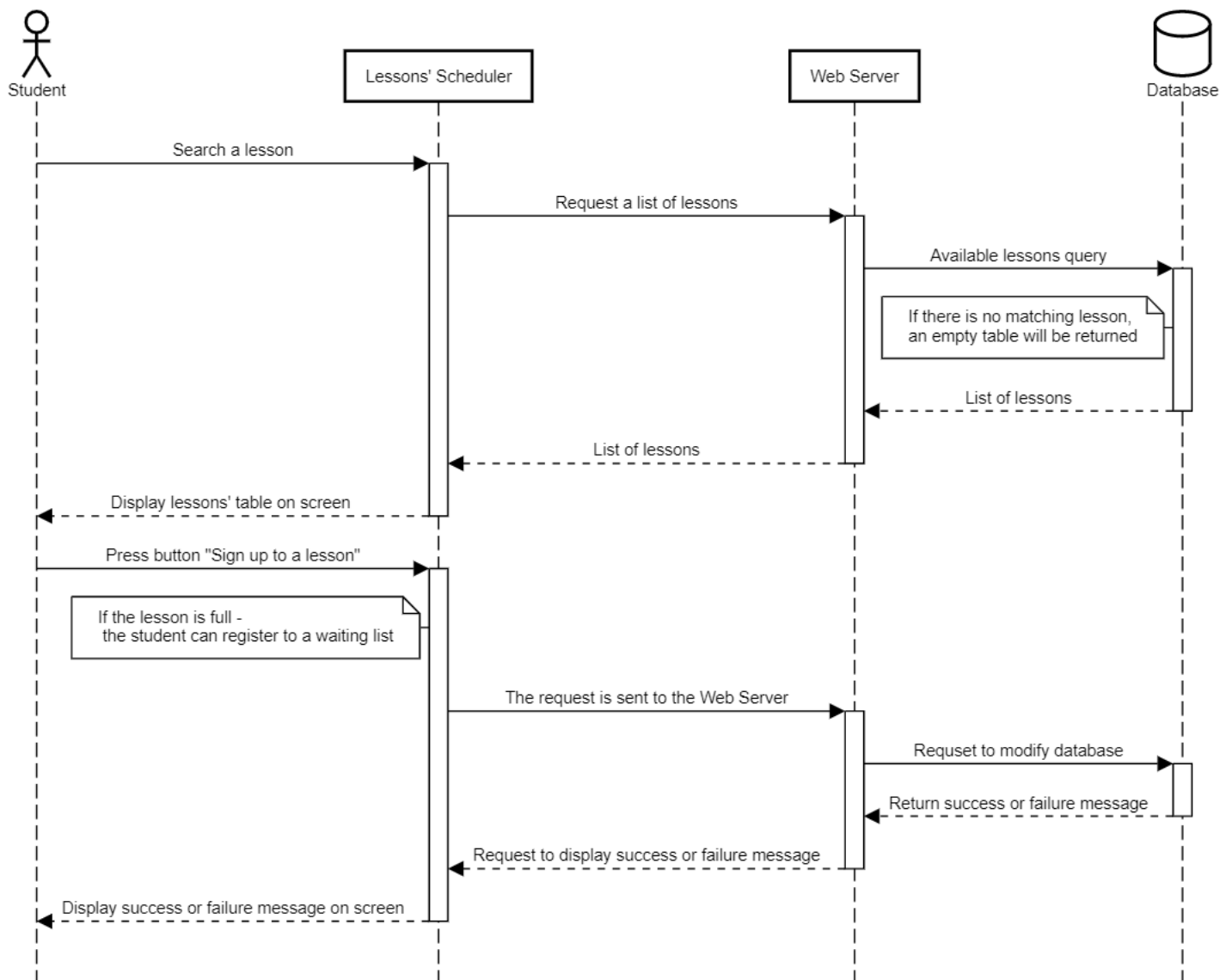


## System Use Case

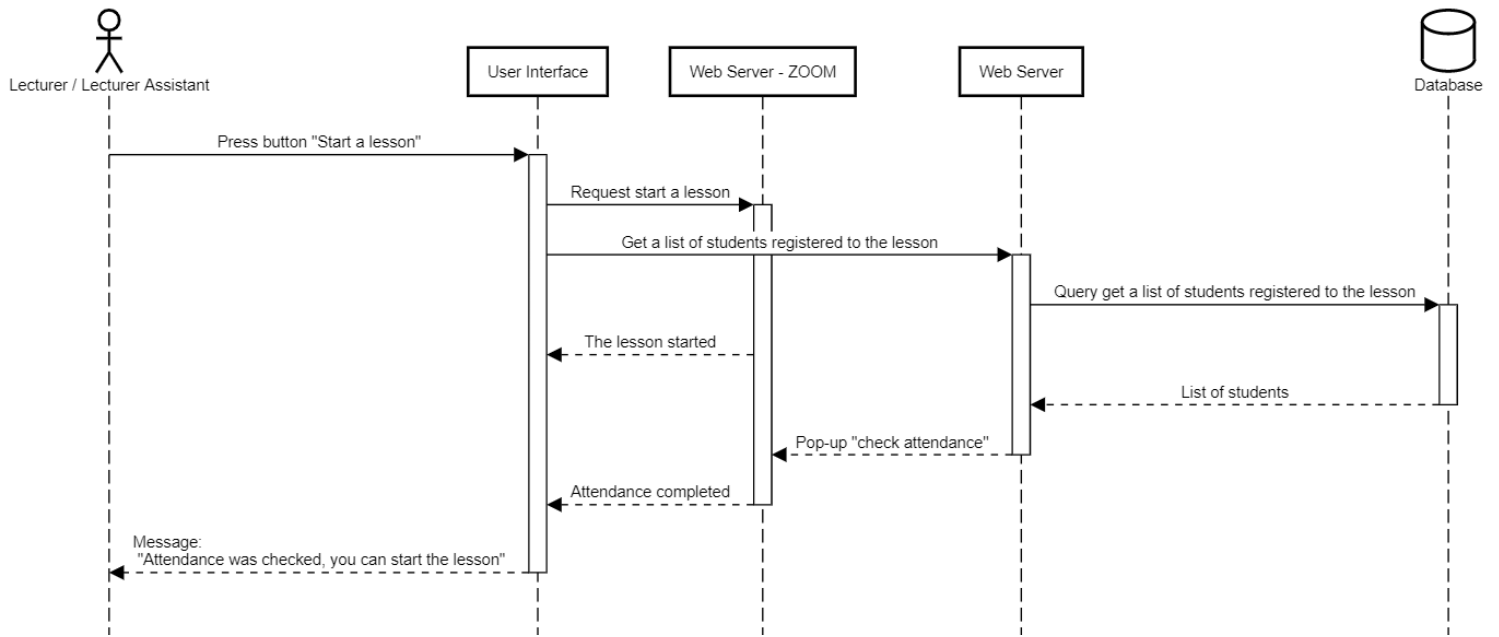




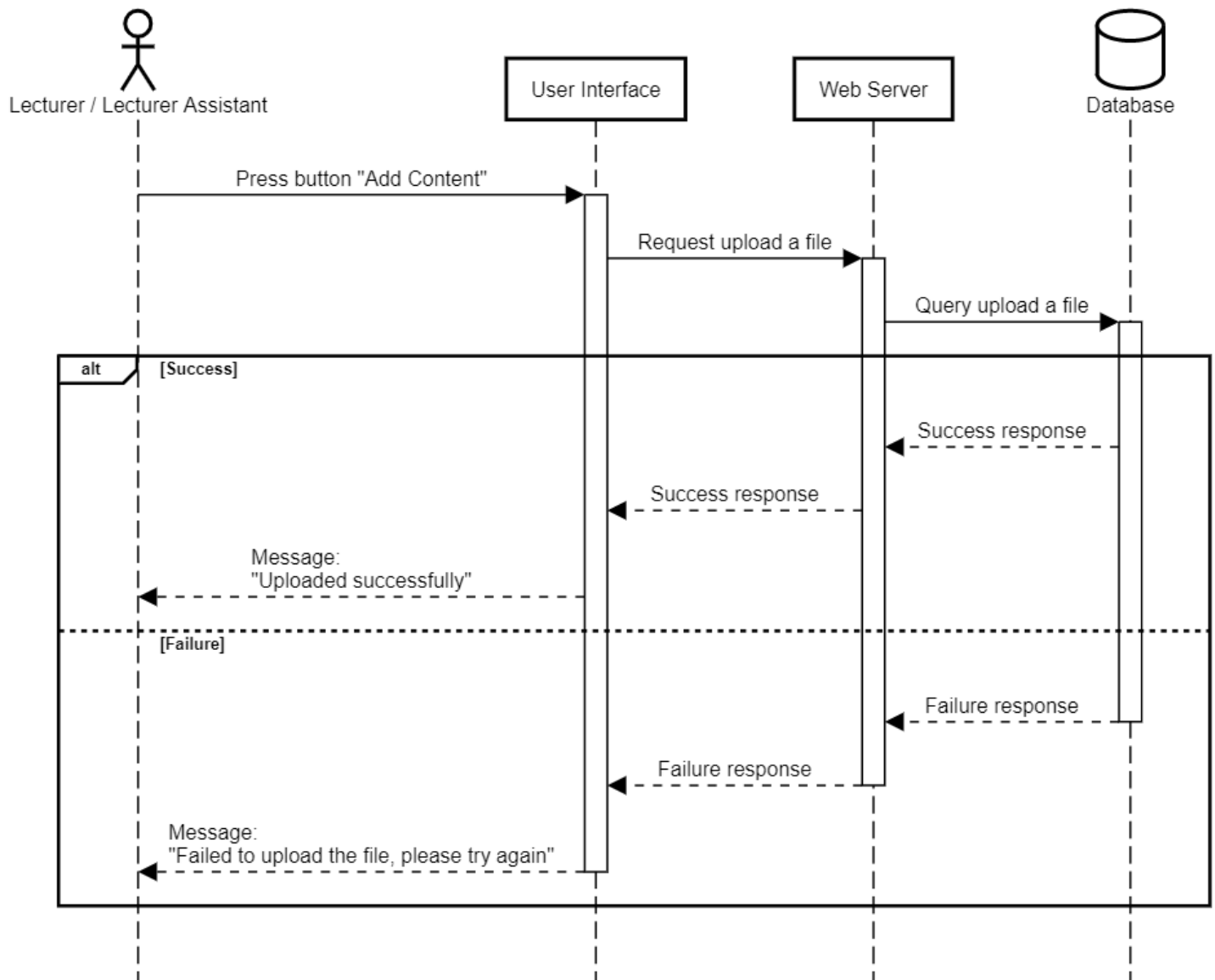
## Find available lessons and sign up to a lesson



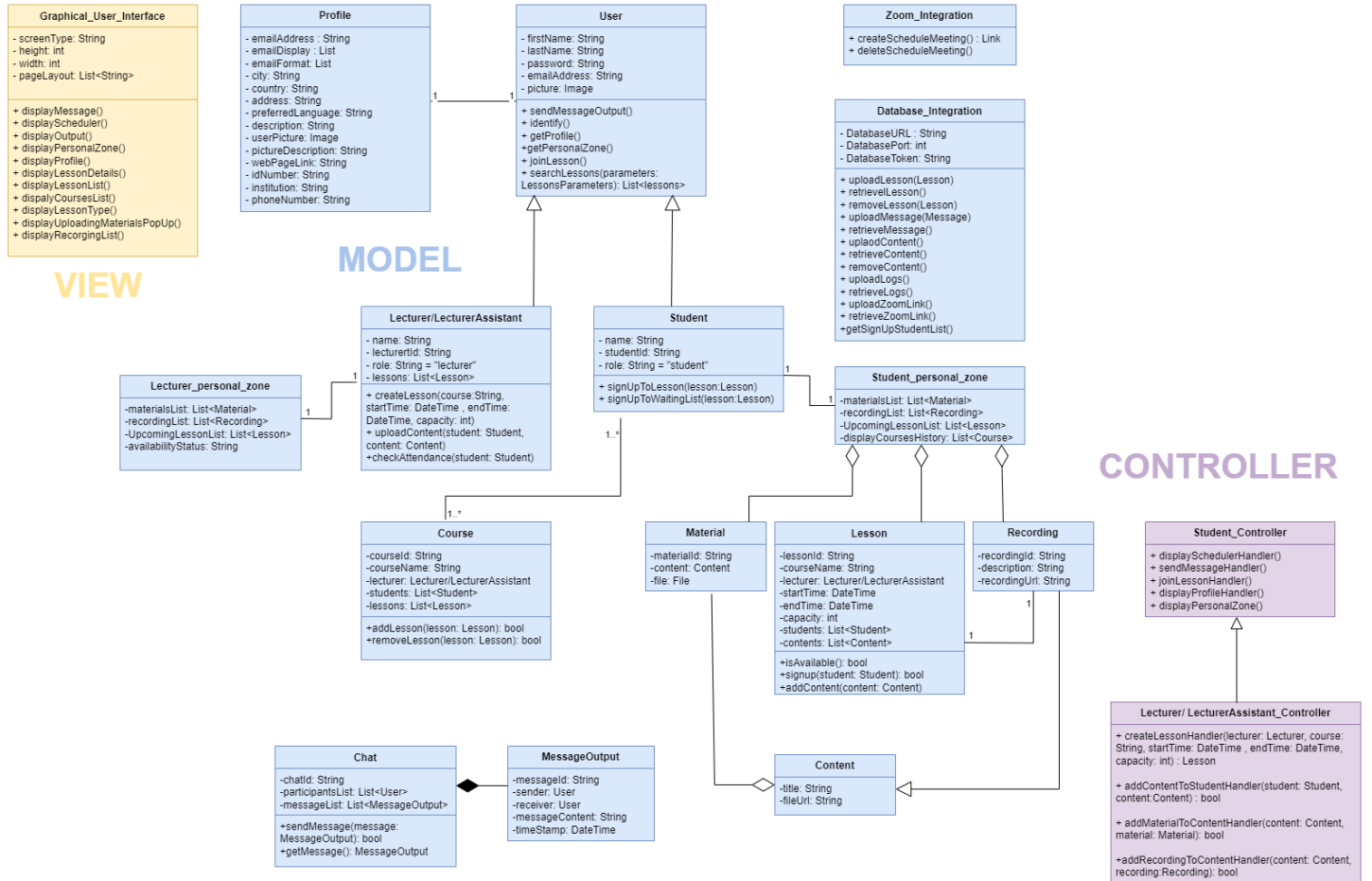
## Start a lesson and check attendance



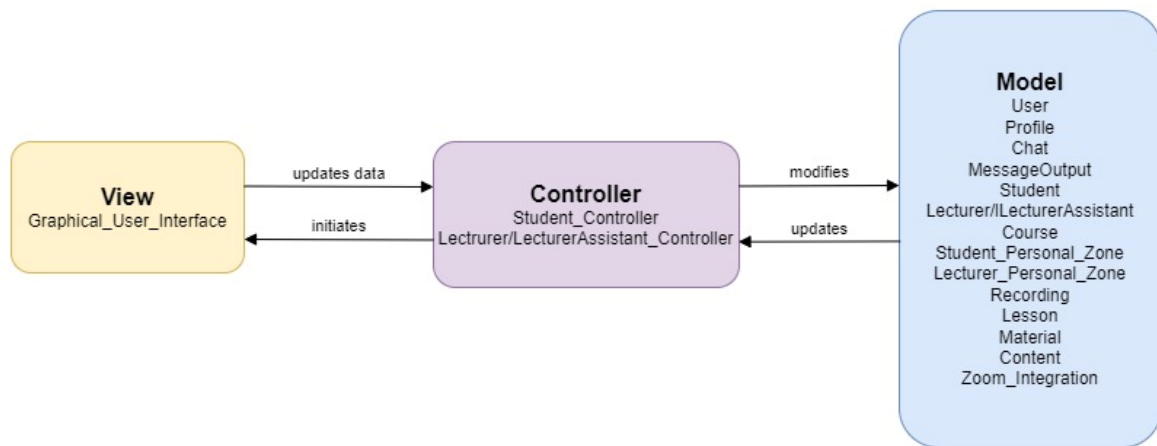
## Upload a file



## Class diagram



## MVC Diagram



## **מבוא:**

בפרויקט זה ניתן לראות יישום של תבנית הארכיטקטורה MVC (Model-View-Controller) המשמשת לפיתוח מערכות אינטרנטיות. מטרתה העיקרית של תבנית זו היא להפריד בין רכיבי המערכת השונים, דבר שמאפשר תחזוקה נוחה יותר, סקלביליות ושימוש חוזר בקוד. הדיאגרמה מחולקת לשלושה חלקים עיקריים: Model, View ו-Controller כאשר לכל אחד מהם יש תחום אחריות ברור ומוגדר במערכת.

## **:Model**

מחלקת ה-Model אחראית על ניהול הנתונים, הלוגיקה ואינטגרציית המערכת. בדיאגרמה, מחלקת ה-Model כוללת מגוון מחלקות כמו: user, lecturer, student, profile ועוד. מחלקות אלה מטפלות בניהול הנתונים והשמירה שלהם, והן כוללות מאפיינים (attributes) ופונקציות לטיפול במידע. לדוגמה, מחלקת ה-User כוללת מאפיינים כמו שם משתמש, סיסמא, מייל, ותמונת פרופיל וכן פונקציות כמו getProfile ו-sendMessageOutput לטיפול בפעולות שקשורות למידע המשתמש.

## **:Controller**

מחלקת ה-Controller מתווכת בין מחלקות ה-Model ל-View. בדיאגרמה, מחלקת ה-Controller כוללת את המחלקות: Student\_Controller ו-Lecturer/LecturerAssistant\_Controller. תפקידן של מחלקות אלה הוא לטפל בפעולות ובבקשות של המשתמשים כגון, פתיחת חשבון, שליחת הודעות ויצירת שיעורים. ה-Controller מקבל בקשות מה-View, מעבד אותן בעזרת ה-Model, ומחזיר את התוצאה ל-View להצגה בפני המשתמש.

## **:View**

מחלקת ה-View אחראית על הצגת המידע למשתמש ועל התקשורת איתו. בדיאגרמה, מחלקת ה-View מיוצגת על ידי ה-GUI (ממשק משתמש גרפי). מחלקה זו כוללת מאפיינים כמו screenHeight, height וכן פונקציות להצגת מידע על המסך כמו displayOutput, displayProfile, displayScheduler. תפקידה של ה-View הוא להיות נקודת הקשר בין המשתמש למערכת, ולאפשר לו לראות ולהגיב למידע המוצג. באמצעות מבנה זה של MVC, המערכת מתפקדת בצורה מובנית, ברורה ויעילה.