

# מסמך תכנון הפרויקט

יוסי כהן

מור פרץ

ימית שפלר



## כלים וטכנולוגיות

### OpenCV

חבילת תוכנה שנועדה לעזור לפתח יישומים של ראייה ממוחשבת. תוכנה במקור ע"י אינטל וכיום נתמכת ע"י קהילת קוד פתוח.

זיהוי חלקי הפנים נעשה ע"י אלגוריתם ויולה ג'ונס כמוסבר בלינק הבא :

[https://docs.opencv.org/3.3.0/d7/d8b/tutorial\\_py\\_face\\_detection.html](https://docs.opencv.org/3.3.0/d7/d8b/tutorial_py_face_detection.html)

דוגמה להרצת OpenCV ע"י העלאת תמונה :



הסיבות לשימוש בממשק זה לעומת ממשקים אחרים בפרויקט :

1. תמיכה רחבה ברשת
2. קוד פתוח
3. ממשק נוח לשימוש

### JPA-Java Persistence API

ספריית Java המאפשרת למפות אובייקטים ב Java לבסיס נתונים. JPA מבוסס על annotations (פירושים) – תגיות Java שמתחילות ב @, ומופיעות לפני הגדרות של class ו member. למעשה דרך ממשק זה אנו יכולים להגדיר ישויות שכל אחת באופן אוטומטי יוצרת טבלה ב DB.

הסיבות לשימוש בממשק זה לעומת ממשקים אחרים בפרויקט :

1. ממשק נוח ונגיש המאפשר גישה ויצירת ישויות ל-DB
2. ישנם מדריכים רבים ברחבי האינטרנט ובנוסף מצגת מפורטת מקורס JAVA.
3. המלצה ע"י המנחה לשימוש בממשק זה.

מסד נתונים יחסי, רב נימי ורב משתמשים מבוסס שפת SQL

במודל זה בסיס הנתונים בנוי מטבלאות, כאשר כל טבלה מכילה מידע על ישות מסוימת, ולכל רשומה בטבלה יש שדה ID שמזהה באופן ייחודי את הרשומה (Prime-key). הקשרים בין הרשומות בטבלאות השונות נעשה באמצעות שדה מפתח, שבו ערכים זהים מסמלים קשר בין הרשומות. שליפת מידע ופעולות עדכון בבסיס נתונים טבלאי נעשות באמצעות שפת SQL, המהווה ממשיק המאפשר גישה לנתונים מבלי להתייחס לאופן שמירתם בבסיס הנתונים.

הסיבות שנשתמש בממשק זה לעומת ממשקים אחרים:

1. MySql הוא בסיס נתונים קל ללימוד ולשימוש באופן יחסי לבסיסי נתונים אחרים.
2. תומך בנפחים גדולים של מידע.
3. תוכנה חינמית.
4. ישנם מדריכים רבים ברחבי האינטרנט שניתן להיעזר בהם.
5. תמיכה של שרתי המכללה.

## תכנון מודולים

### צד לקוח



הסבר לפי Package :

• Screen

• Package login

- **LoginScreen** - מחלקה המתארת את דף הכניסה של המערכת.
- **GuestScreen** – מחלקה המתארת את הדף שבו אורח לא רשום מתעניין במערכת. בדף זה יהיה סרטון הדרכה על המערכת והשאר פרטים.

• Package SchoolAdmin

- **AdministrativeMainMenu** – מחלקה המתארת את הדף הראשי שמנהל אדמיניסטרטיבי נכנס למערכת. בפניו אפשרות לבחור רשימת משתמשים או הנפקת דו"ח חריגות.
- **AllUsersScreen** – מחלקה המתארת את הדף המייצג את רשימת כל המשתמשים במערכת, סטודנטים ומורים.
- **AddUserScreen** – מחלקה המתארת את הדף שבו ניתן להוסיף משתמש חדש למערכת, מורה או תלמיד.
- **ExceptionScreen** – מחלקה המתארת את הדף המציג הנפקת דו"ח חריגות. פרטים על סטודנט, פרטים על השיעור, סוג החריגה (מבט או מעקב עמוד) וזמן החריגה.

• Package Student

- **ScheduleScreen** - מחלקה המתארת את הדף שבו תלמיד נכנס למערכת והוא לא מחובר לשיעור. דף זה יציג את מערכת השעות של התלמיד.
- **StudentLessonScreen** – מחלקה המתארת את הדף שבו תלמיד מחובר למערכת בזמן שיעור. מה שיוצג לתלמיד זה תוכן השיעור שהמורה העלה.
- **StudentQuestionPopup** - מחלקה המתארת את הדף (popup) שקופץ למסך של תלמיד בזמן שיעור כאשר המורה מציג שאלה.

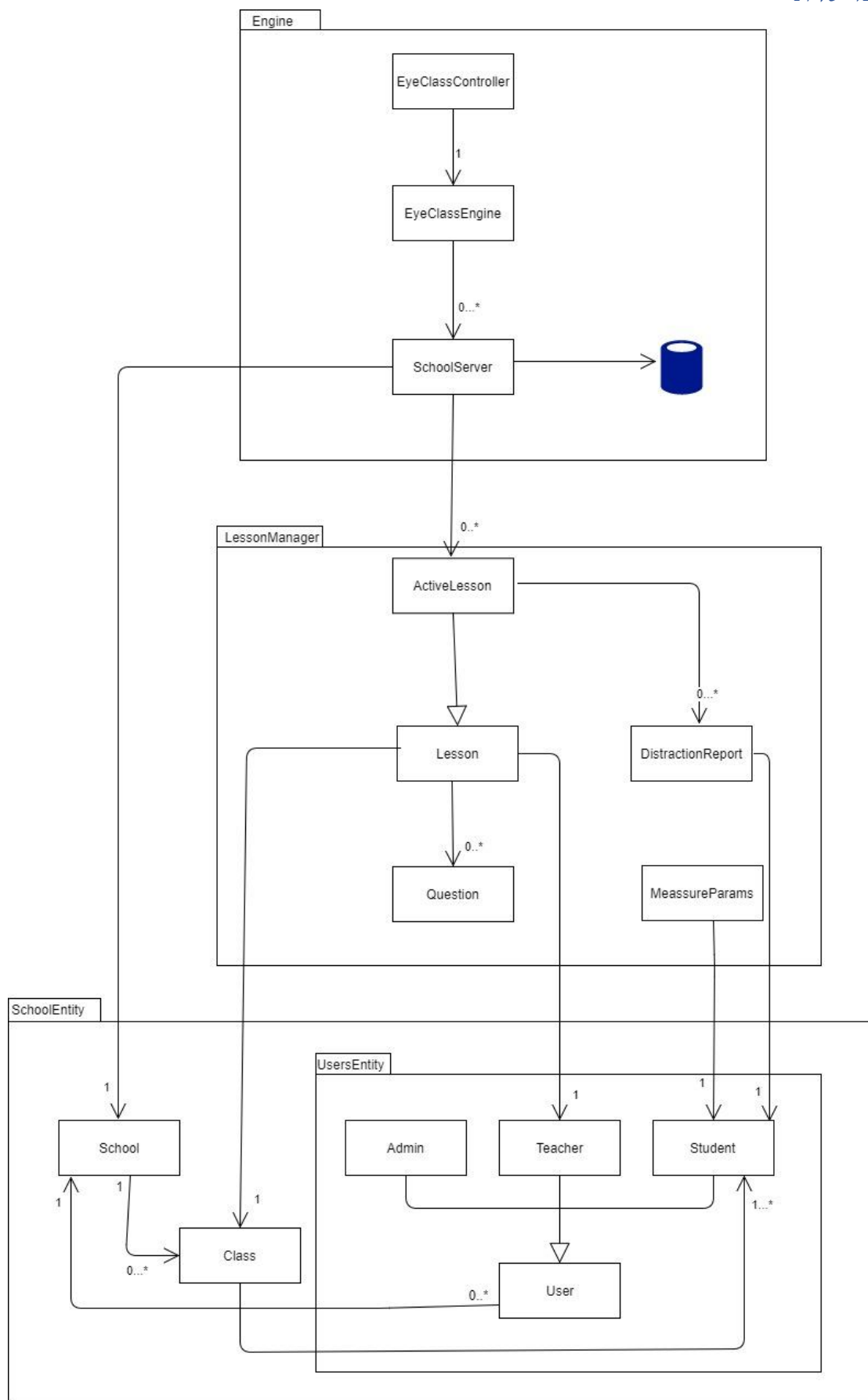
• Package Teacher

- **TeacherMainMenu** - מחלקה המתארת את הדף הראשי שבו מורה נכנס למערכת, הוא יוכל לבחור בין הוספת מערך שיעור, בחירת מערך שיעור והוצאת דו"חות חריגות.
- **AddLessonScreen** – מחלקה המתארת את הדף שבו מורה יכול להוסיף מערך שיעור למערכת.
- **AddQuestionScreen** - מחלקה המתארת את הדף שבו מורה מוסיף שאלה למערך שיעור.
- **SelectLessonScreen** - מחלקה המתארת את הדף שבו מורה בוחר מערך שיעור.
- **TeacherLessonScreen** - מחלקה המתארת את הדף שמורה רואה בזמן שיעור, דף זה מכיל את התוכן של מערך השיעור, הצגת השאלות הקשורות לשיעור, ורשימת התלמידים המסומנים ע"פ הקשב שלהם (מעקב עיני או מעקב דף).
- **TeacherQuestionPopup** - מחלקה המתארת את הדף (popup) שקופץ למורה כאשר הוא בוחר שאלה להציג בפני התלמידים.
- **TeacherExceptionScreen** – מחלקה המתארת את הדף המייצג דו"ח חריגות של התלמידים ע"פ שיעור מסויים.
- **TeacherQuestionReportScreen** - מחלקה המתארת את הדף המייצג דו"ח של מענה תלמידים לשאלות שנשאלו בכיתה ע"פ שיעור מסויים.

• Tracking

- **StudentTracking** - מחלקה המקבלת מידע אודות מצב תלמיד בזמן שיעור. במחלקה זאת אנו דוגמים את הדף הנוכחי שבו התלמיד מסתכל בזמן שיעור, וכמו כן האם הוא מסתכל למסך.

- **CommunicatorManager** - מחלקה המקשרת בין צד השרת לצד הלקוח. דרך מחלקה זאת אנו מקבלים מידע מהשרת וכמו כן שולחים מידע לשרת אודות מצב תלמיד בזמן שיעור.

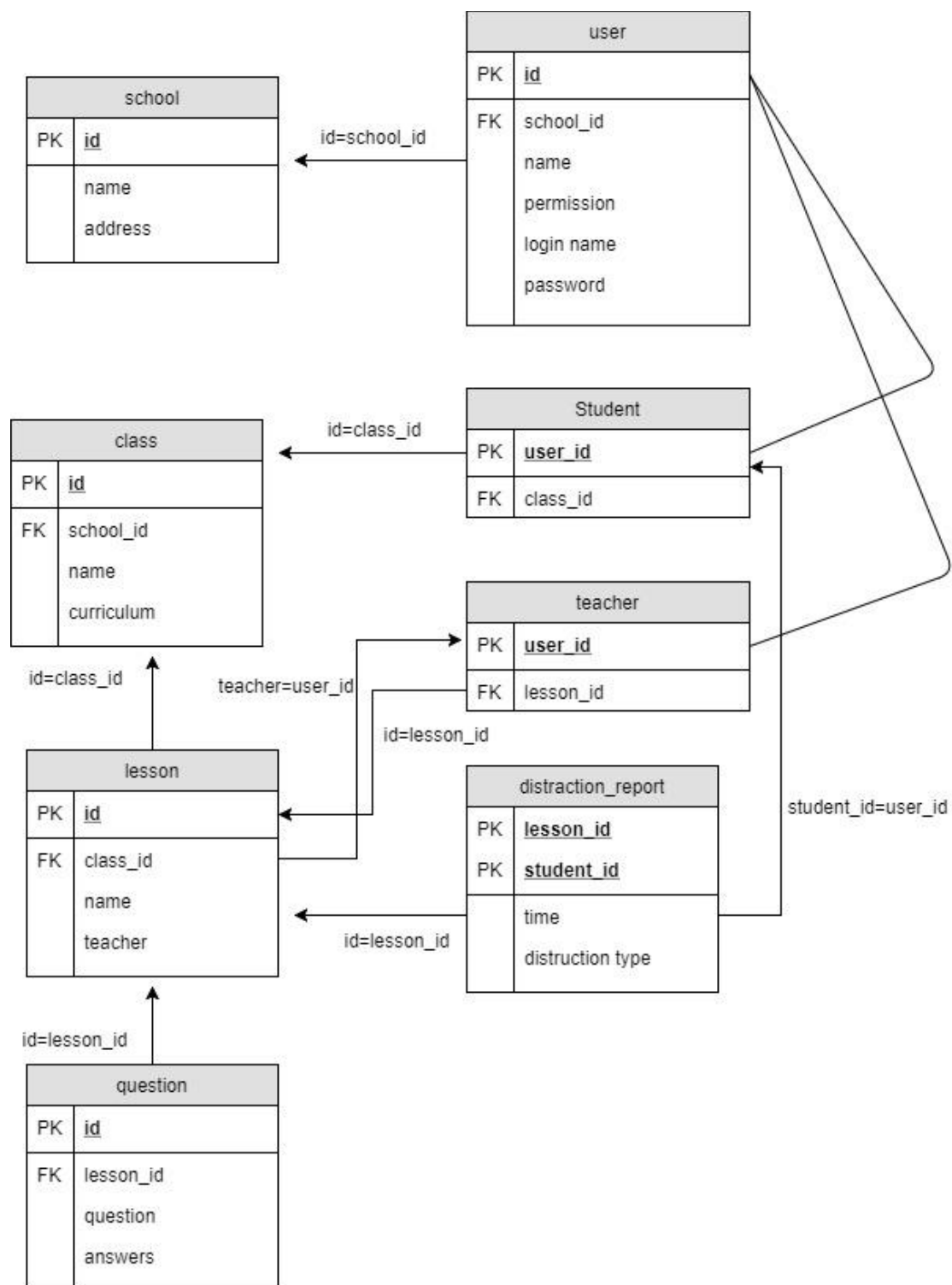


הסבר לפי Packages :

- **SchoolEntity** – ישויות בבתי ספר, משמש למטרת "בניית" ביה"ס ונתונים גולמיים לשמירה בבסיס נתונים.
- **LessonManager** – משמש עבור בניית מערכי שיעור והפעלת שיעור בפועל וניהולו בזמן אמת.
  - **Lesson & Question** – מחלקות המשמשות לבניית מערך שיעור.
  - **ActiveLesson** – אובייקט שנבנה בעת שהמרצה בוחר טען שיעור (קורה בתחילת השיעור הפרונטלי). מרכז את המידע על אותו שיעור ספציפי אל מול אותה כיתה בזמן אמת. נועד כדי לרכז את דו"חות החריגות בזמן אמת, מכיל את מספר העמוד של המרצה.
  - **MeasureParams** – נוצרת בזמן שיעור כדי למדוד את מצב התלמיד, מקבלת מידע מצד לקוח שמכיל את הצילום של התלמיד ומספר העמוד, משווה את מספר העמוד אל מול מספר העמוד של המרצה ובודק האם התלמיד ממוקד במסך בעזרת OpenCV.
  - **DistractionReport** – אובייקט שנוצר רק אם זוהתה חריגה, יכול בנוסף את המידע הגולמי הבא:
    - תאריך
    - משך זמן החריגה
    - סוג החריגה(מבט / מספר עמוד לא נכון)
- יישמר ב-ActiveLesson עד לסיום השיעור על מנת לשרשר חריגות זהות(למשל אם התלמיד לא עקב אחר מספר העמוד במשך 10 דגימות נרצה לשמור חריגה אחת ולא 10 חריגות).
- בסוף השיעור יישמר במסד הנתונים.
- **Engine** – תקשורת בין צד שרת לצד לקוח. עבור כל ביה"ס ייפתח אובייקט של SchoolServer שירכז את הפעילויות של ביה"ס לשיעורים פעילים וגישות למסד הנתונים עבור שאילות לחריגות.



## בסיס נתונים



## שילבי התקדמות

- **שלב א':** תאריך סיום: 20.3.18
  - בניית שרת שיתמוך בפעולות הבאות ע"י Junit:
    - הוספת מערכי שיעור
    - הוספת משתמשים (תלמיד ומרצה לשלב ראשון)
    - שליפת מערכי שיעור
  - בניית בסיס נתונים עם תמיכה במשתמשים ומערכי שיעור.
- **שלב ב':** תאריך סיום: 5.4.18
  - בניית המסכים הבאים (זמן שיעור בלבד):
    - השיעור יבחר באופן "Hard-coded" ע"י השרת.
    - מסך הצגת מערך שיעור למרצה הכולל דפדוף בקובץ PDF.
    - מסך הצגת שיעור לתלמיד הכולל דפדוף בקובץ PDF ונעילת כפתורי הבית.
  - בדיקת מספר עמוד אצל המרצה והתלמיד ושליחתם לשרת (ללא עיבוד ויצירת חריגות)