



# מסמך ארכיטקטורה [MultiLive] – תש"פ

**צוות:** ניר יוסף ודניאל זיגלר שמואל

**כיתה:** אשקלון 11

**ראש צוות:** שם ראש הצוות

**מנטור:** שם המנטור

### הסבר כללי על המסמך

ברכותינו! סיימתם לתכנן את עיקר "הסיפור העסקי" של הפרויקט, עוד רגע יגיע הזמן להתחיל לממש... המסמכים עליהם עבדתם עד כה עסקו בתוכן של הפרויקט שלכם - מה הוא עושה. כעת הגיע הזמן להבין ולתכנן כיצד המימוש יתבצע.

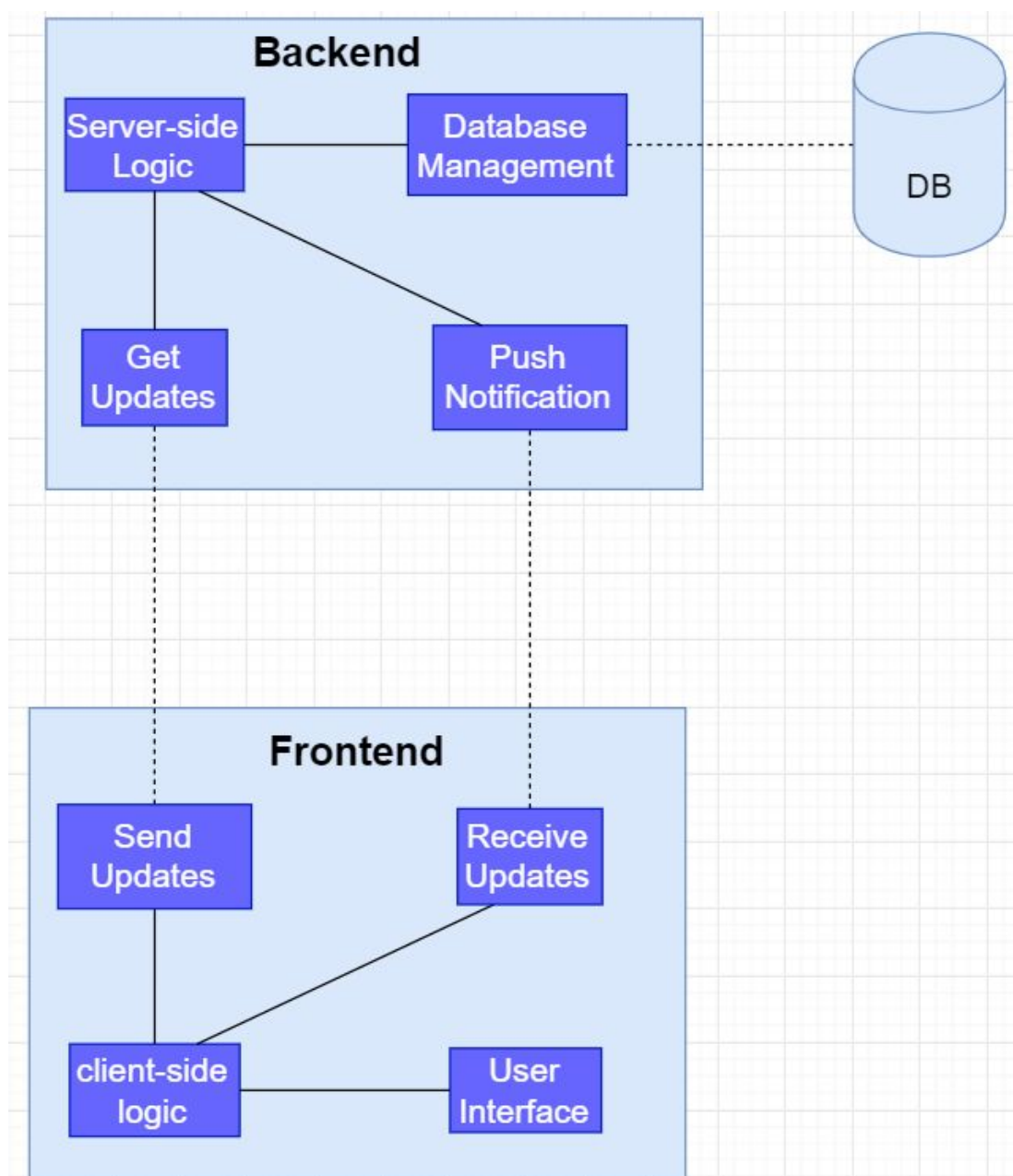
מטרת מסמך הארכיטקטורה היא להבנות את השלד העיקרי למימוש הפרויקט שלכם - האופן שבו הוא יבנה, הרכיבים השונים, חלוקת האחריות והקשרים ביניהם, ועוד.

תכנון נכון של הפרויקט, באמצעות מסמך זה, יסייע לכם להמנע מטעויות בזמן הפיתוח, וגם יאפשר לחלק את העבודה על הפרויקט בין חברי הצוות וכן לחלקה למספר ספרינטים/איטרציות - כאשר בכל איטרציה תתמקדו במספר מצומצם של רכיבים או חלקים מתוך כלל הפרויקט.

במסמך זה נתעסק בתמונה הגדולה, ולאחר מכן מדי 3 שבועות תפתחו ב"ספרינט" חלק ספציפי מתוך הפרויקט.

## מבט על

כאן יש לפרט את הרכיבים העיקריים שמרכיבים את מימוש הפרויקט. אין צורך להכנס לדקויות או למקרי קצה, אלא לתת תמונה כוללת של החלקים העיקריים במערכת. ראשית כל יש לשים כאן **דיאגרמה** המתארת את הרכיבים העיקריים והקשרים ביניהם. את הדיאגרמה ניתן ועד לשירותי רשת Powerpoint ליצור במגוון דרכים - החל מסריקה של שרטוט שציירתם בעצמכם, דרך Draw.io או moqups.com חינמיים כמו לאחר מכן יש **להסביר** בקצרה עבור כל רכיב לגבי תפקידו ותחום האחריות שלו, לאילו רכיבים אחרים הוא במסמכים הקודמים (כלי <sup>1</sup> מתממשק. לדוגמה, חלק ממסמך הארכיטקטורה מהפרויקט שממוספר במספר הבודק חורי אבטחה ברשתות מחשבים פנימיות).



## הסבר על כל רכיב ותפקידו:

● **Backend** – מטרתו היא לטפל בכל הבקשות ולהיות השלד של הפרויקט. החלק הזה אחראי על ביצוע כל הפעולות שקשורות לקישור בין משתמשים למסד הנתונים וקישור בין משתמש אחד לאחר. הBackend מתקשר עם הלקוח בעזרת socket כאשר הוא גם מאזין ללקוח וגם שולח לו הודעות ועדכונים.

● **Frontend** – החלק הזה הוא בעצם מה שהמשתמש רואה כאשר הוא נכנס לאתר והוא מכיל את web gui. גם בחלק הזה מתבצעות פעולות כמו קישור אל השרת והרצת פקודות לפי בקשת המשתמש.

● **DB** – מסד הנתונים של הפרויקט, מכיל מידע רב כמו משתמשים נתונים לגבי האתר וקבצים שנשמרו.

## עיצוב הנתונים ויישויות מידע

דרך טובה להבין את המערכת היא דרך הנתונים ויישויות המידע שעוברים ונשמרים בה. הכוונה היא למידע שמועבר בתוך המערכת (למשל בין מודולים / רכיבים שונים, או בין לקוח לשרת על קבצי טקסט או בינאריים). שימו לב כי DB- גבי הרשת) וגם למידע שנשמר על ידי המערכת (למשל ביישויות יכולות להכיל יישויות אחרות.

בפרויקט שלנו מידע מועבר מצד השרת לצד הלקוח על מנת לעדכן את המשתמש במידע האישי ופרטים נוספים שלא דורשים שמירה, לעומת זאת המידע המועבר בין צד הלקוח לצד השרת הוא בקשות הדורשות שמירה במבנה הנתונים.

בקשות הנשלחות על ידי הלקוח:

- הרשמה או שינוי נתונים אישיים.
- שינוי שנעשה באובייקטים המוצגים על canvas.
- שינוי שנעשה על אובייקט.

מידע המועבר מהשרת אל הלקוח:

- נתונים אישיים של המשתמש.
- תגובה לבקשת התחברות או הרשמה.
- שינויים שנערכו בקובץ על ידי משתמשים אחרים.
- קבצי קול (voip)

## טכנולוגיות עיקריות

בהתבסס על הפירוט מהסעיפים הקודמים במסמך זה, ציינו כאן במרכז את כל הטכנולוגיות שהוזכרו במסמך. רשימה זו צריכה לכלול שפות פיתוח, רכיבים ושירותים חיצוניים, ממשקים מיוחדים (כמו מערכות הפעלה, חומרה וכו') במידה ויש כאלו. מבין כל אלו, האם ישנם נושאים בהם אתם מרגישים שאינכם מנוסים מספיק (או אפילו לא מכירים בכלל)? מדוע החלטתם להשתמש דווקא בטכנולוגיות האלו? אילו אפשרויות נוספות פסלתם ומדוע (מהן היתרונות והחסרונות שלהן לעומת הטכנולוגיות שבחרתם בסופו של דבר)?

**הטכנולוגיות בהן נשתמש עבור כל רכיב בפרויקט שלנו:**

- צד השרת יעשה בעזרת `django` ויכתב בשפת `python`.
- צד הלקוח יעשה בעזרת `angular` ובשפות `html`, `css`, `typescript`.
- מסד הנתונים יכתב בשפה `MySQL/sqlite`.
- הקישור בין צד הלקוח לצד השרת יעשה בעזרת `socket`ים של האזנה ושליחה.
- השיחה הקולית תתבצע בעזרת טכנולוגיות `VoIP`.
- מערכת `layer system` בצד הלקוח שתבנה ב-`angular`.

# התאמה לאפיון

עבור כל דרישה מפרק האפיון, ציינו מהם הרכיבים אשר מספקים מענה עבורה. במידה וישנם מספר רכיבים הנובעים מדרישה אחת, יש לציין את סדר הפעולה ביניהם.  
לדוגמה:

<u>פיצ'ר</u>	<u>רכיבים רלוונטים</u>
התחברות ויצירת משתמש	לקוח (web gui < לוג'יק < תקשורת עם השרת) < שרת (לוג'יק < מסד הנתונים < החזרת תשובה)
פתיחה של קובץ קיים או יצירת קובץ חדש	לקוח (web gui < לוג'יק < תקשורת עם השרת) < שרת (לוג'יק < מסד הנתונים < תשובה)
ציור ובתיבה	לקוח (web gui < לוג'יק < תקשורת עם השרת) < שרת (לוג'יק < עדכון שאר המשתמשים < מסד הנתונים < החזרת תשובה)
שיתוף הקובץ עם אנשים	לקוח (web gui < לוג'יק < תקשורת עם השרת) < שרת (לוג'יק < תקשורת עם האנשים < מסד הנתונים < החזרת תשובה)
עריכה בו זמנית	לקוח (web gui < לוג'יק < תקשורת עם השרת) < שרת (לוג'יק < עדכון שאר המשתמשים < מסד הנתונים < החזרת תשובה)
היסטורית עריכות	לקוח (web gui < לוג'יק < החזרה למצב הקודם בעזרת המידע שנשמר על האתר)
שיחה קולית עם הנוכחים בקובץ	לקוח (web gui < לוג'יק < תקשורת עם השרת) < שרת (לוג'יק < עדכון שאר המשתמשים < החזרת תשובה)
שמירת הקובץ	לקוח (web gui < לוג'יק < תקשורת עם השרת) < שרת (לוג'יק < מסד הנתונים < החזרת תשובה)
הדפסת הקובץ	לקוח (web gui < לוג'יק < יצירת תמונה < הדפסה)