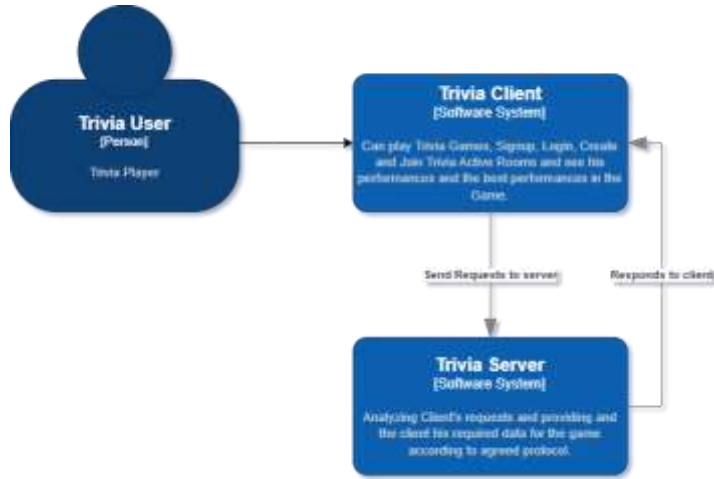


מסמכי עיצוב – "פרויקט הטריווייה" מגישים: עידו רדיד ושומ טורם

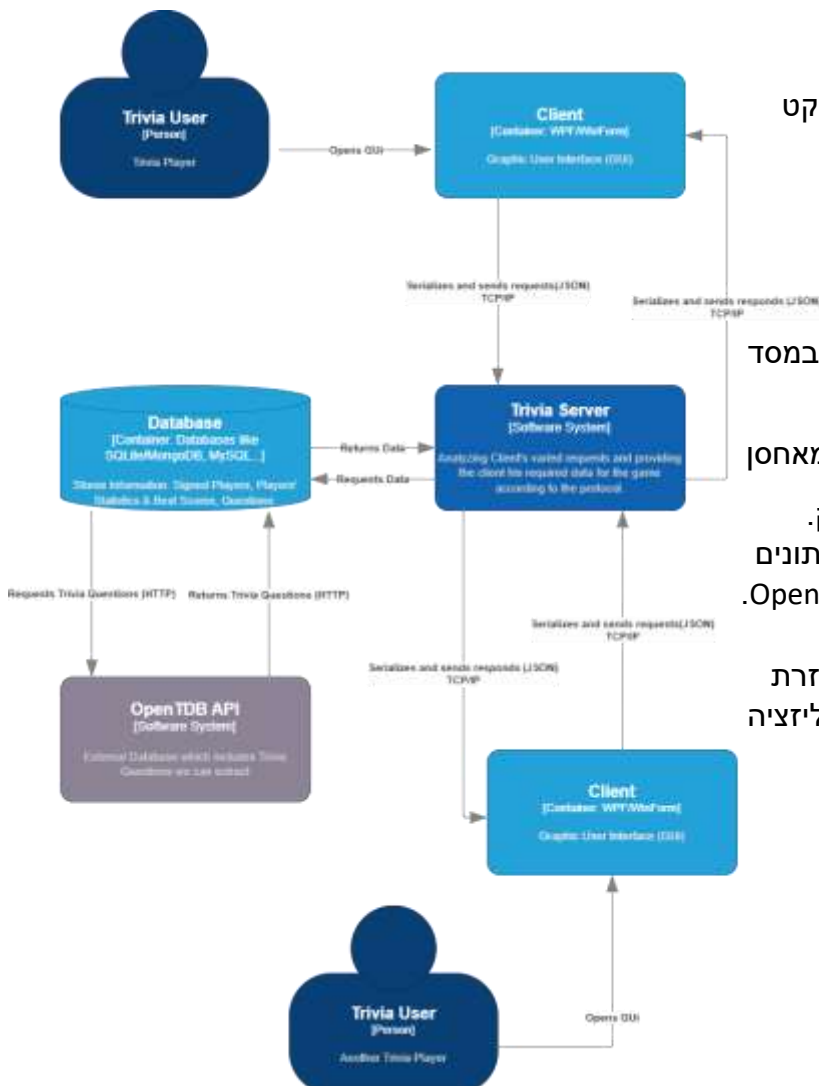
תרשים 1 – High Level Design



בפרויקט הטריווייה קיימים 3 רכיבים מרכזיים –
המשתמש – אנחנו בני האדם
הלקוח – הרכיב הטכני שמייצג אותנו השחקנים
במשחק. רכיב זה מתקשר אל מול השרת.
השרת – הרכיב המרכזי שמנהל את המשחק,
שמנתח את ההודעות של הלקוח, מבצע פעולות
מאחורי הקלעים.

הלקוח והשרת מנהלים תקשורת על ידי פרוטוקול מוסכם ביניהם בצורה מחזורית כל עוד החיבור ביניהם תקף.

תרשים 2 – Container Diagram



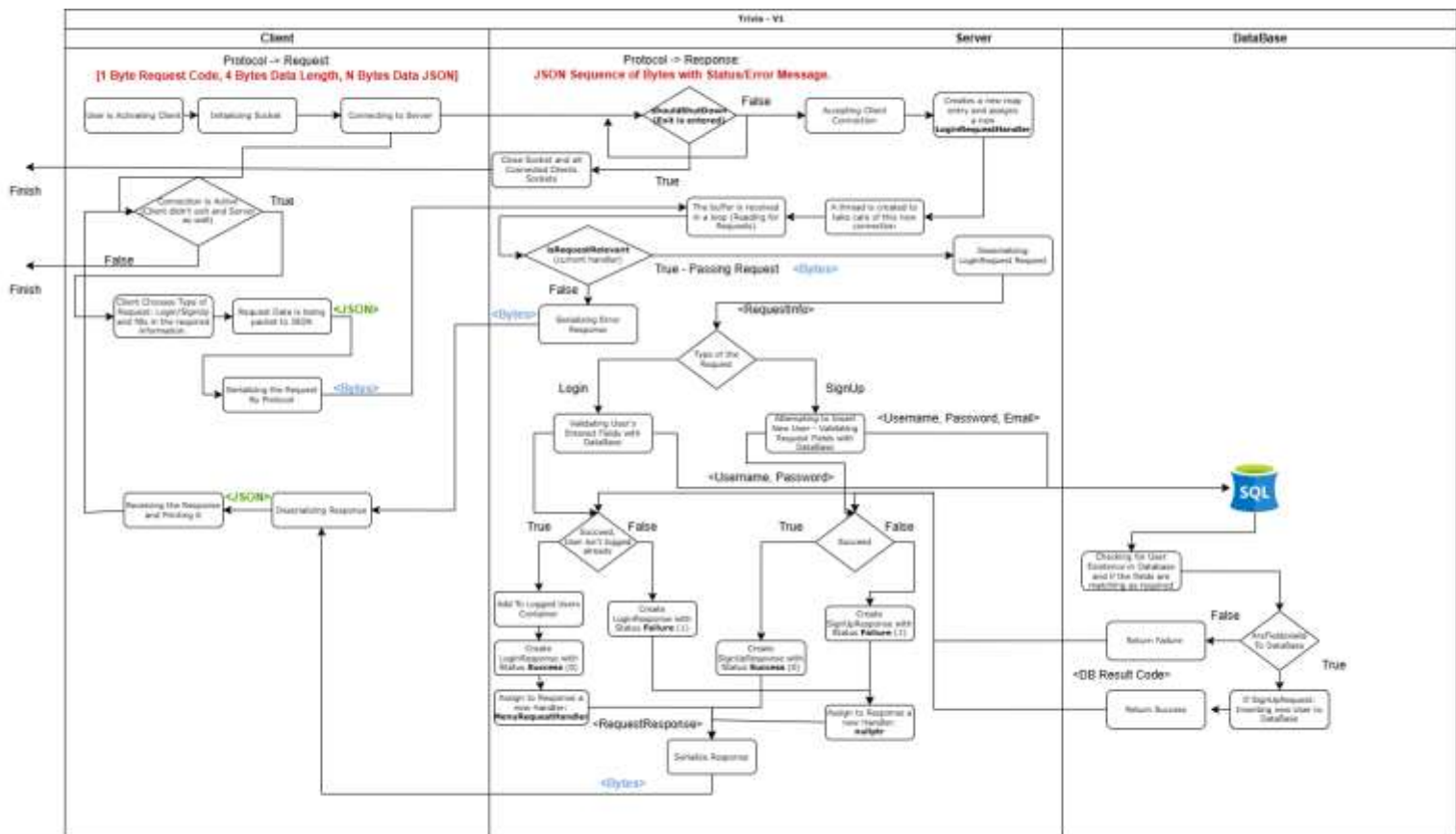
מסמך זה צולל עומק נוסף אל תוך רכיבי הפרויקט המרכזיים.

המשתמשים השונים פותחים את ה-Client שלהם המיוצג על ידי ממשק גרפי GUI. בנוסף לכך, ניתן לראות כי השרת מסוגל לתקשר עם כמה לקוחות בו זמנית. הוא נעזר במסד נתונים אשר מספק מידע אודות המשתמשים ואודות סטטיסטיקות, נתונים אישיים ואף הוא מאחסן שאלות טריווייה שנעזרים בהם במהלך המשחק.

מסד הנתונים עצמו אף מתקשר אל מול מסד נתונים חיצוני באמצעות HTTP, במקרה שלנו OpenTDB API. שאלות טריווייה שנעזרים בהם במהלך המשחק. מסד הנתונים עצמו אף מתקשר אל מול מסד נתונים חיצוני באמצעות HTTP, במקרה שלנו OpenTDB API.

התקשורת בין השרת לבין הלקוחות מבוצע בעזרת TCP/IP ומתקשר על ידי סריאליזציה ודה-סריאליזציה של הודעות JSON.

תרשים 3 – Sequence Diagram -



נקודות חשובות -

1. הלקוח והשרת מתקשרים בעזרת פרוטוקול בינארי בפורמט הבא:
 Headers – בית ראשון הוא קוד ההודעה, לאחריו 4 בתים שמייצגים את אורך ההודעה (N) של ה-Data.
 JSON – Data באורך N בתים הכולל את הבקשות השונות של הלקוח אל מול השרת.
 על מנת לפענח את התגובות והבקשות גם השרת וגם הלקוח מבצעים סריאליזציה ודה-סריאליזציה של ההודעה על מנת להמיר את מה שהם שולחים לבתים או לתרגם את מה שהם מקבלים ל-JSON או מחלקה שמייצגת אותו.
2. על מנת לסיים את התקשורת הלקוח יכול לבחור לצאת מבין תפריט האפשרויות שלו. בשרת עצמו פשוט צריך להכניס ל-Console 'EXIT' ולאחר מכן השרת ינתק את כל הלקוחות המחוברים. לאחר סיום התקשורת של השרת התוכנית מסתיימת, לאחר סיום הניתוק של הלקוח, השרת נשאר לרוץ על מנת לטפל בלקוחות האחרים.
3. בסיום הבדיקות של השרת ועדכון הנתונים במסד הנתונים בהתאם לשדות שהוכנסו בבקשת המשתמש (ב-JSON שהוא שלח כ-Data), השרת מעדכן את מצב הלקוח בהתאם (אם מדובר בשגיאה אז המצב שלו יהיה מיוצג כ-nullptr ואם מדובר בהתחברות מוצלחת אז המצב שלו יתחלף למצב הבא MenuRequestHandler כך שהוא יוכל לראות ולגשת בתפריט. בנוסף השרת אף מחזיר ללקוח סטטוס הודעה המייצג אם הבקשה שלו עברה בהצלחה או לא.