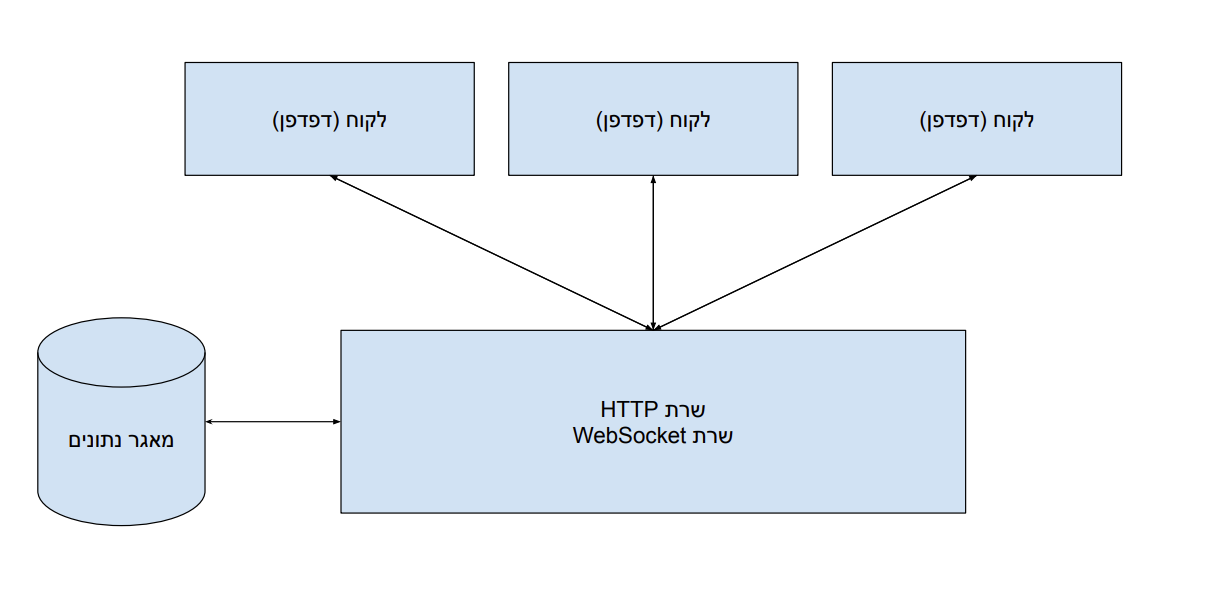
# אורי פלייש ראיון עבודה

## פרויקט

הפרויקט הוא חיקוי למשחק [skribbl](https://skribbl.io/), משחק למספר אנשים בו כל משתתף בתורו מצייר ציור ועל יתר המשתתפים לנחש מהו הציור שהשחקן שמצייר צייר.

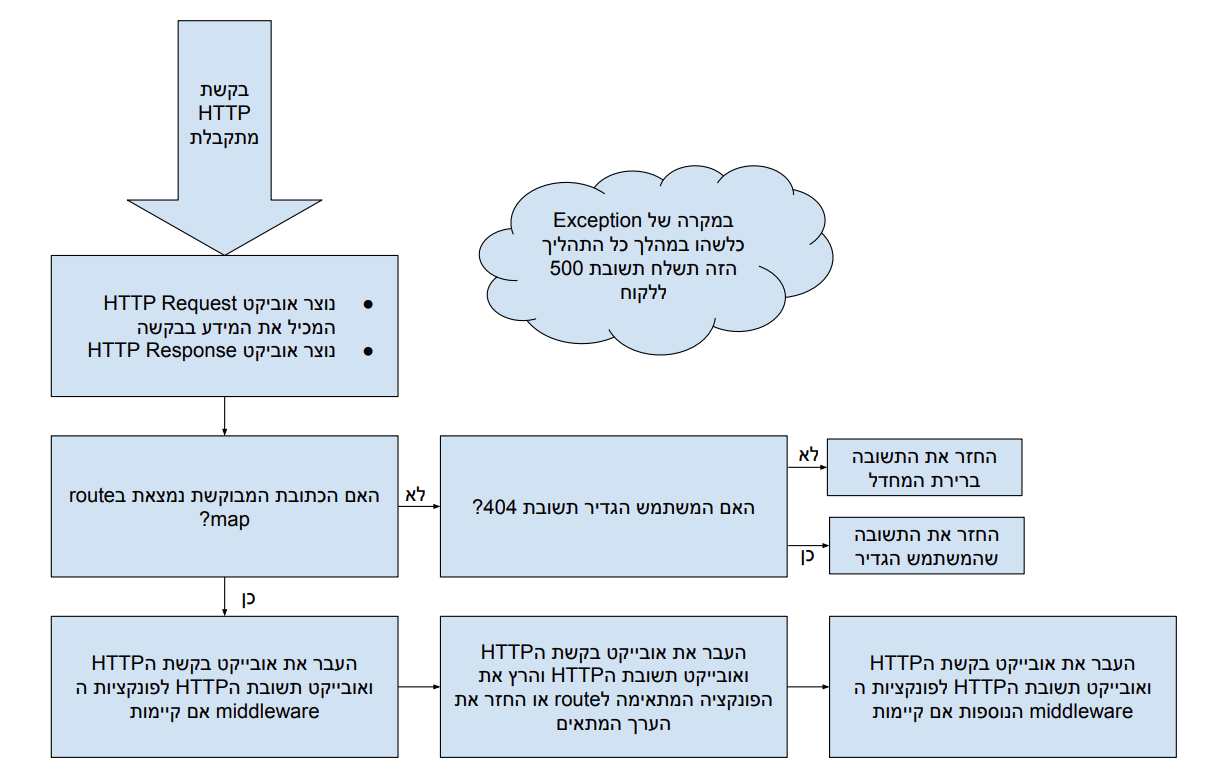
הפרויקט מורכב מ3 מערכות עיקריות.

* שרת HTTP.
* שרת Web Socket האחראי גם על הלוגיקה הפנימית של המשחק.
* מערכת Authentication האחראית לאימות משתמש ותקשורת עם מאגר הנתונים.

### שרת הHTTP

שרת הHTTP בנוי כך שלא ישרת רק בפרויקט הספציפי הזה, אלא בתור ספריה שאנשים יוכלו להשתמש בה בפרויקטים שלהם לשימושים שונים בתכלית ושאנשים יוכלו לכתוב תתי-ספריות עבור הספרייה עצמה. מדובר בספרייה low-level המאפשרת למשתמש להגדיר routes משלו ולהתעסק ישירות עם מבנה תשובת ובקשת הHTTP. הספרייה מאפשרת גם להקצות תיקיות שלומות לroutes ומאפשרת הגדרת פונקציות middleware בדומה לספריית express של node.js.

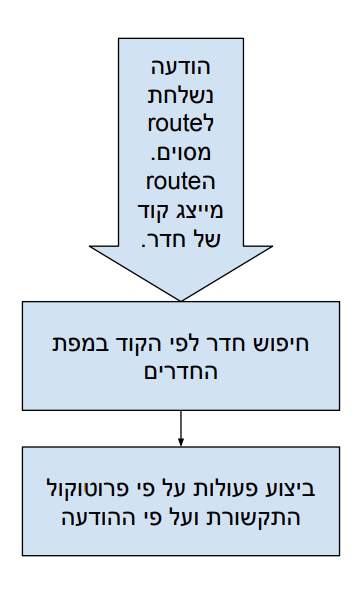
#### תרשים מערכת שרת הHTTP



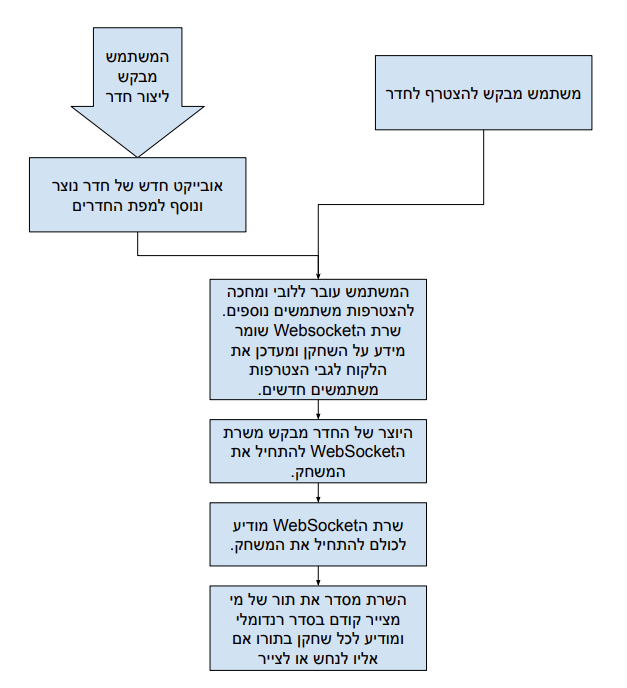
שרת הWeb Socket

שרת הWeb Socket אחראי לתקשורת במהלך המשחק ובמערכת זאת כתובה הלוגיקה הפנימית של המשחק. לצורכי אימות ביצוע רוב הפעולות נדרש שליחה של הsession ID של המשתמש, פירוט על כך בהמשך.

אופן התקשורת מתבצע באופן הבא:



*יצירת חדרים וכניסה לחדרים*



*ציור וסינכרון הציור בין משתמשים*

כאשר משתמש מבצע פעולות שונות הקשורות לציור הוא שולח אותם באמצעות הWeb Socket לשרת. השרת לאחר מכן שולח את הפעולות לשאר הלקוחות. הלקוחות חוזרים על הפעולות בקנבס שלהם ובכך הציור מסתנכרן. פעולות כמו לחיצה על העכבר, שחרורו ושינוי צבע.

*שליחת הודעות וניחושים*

בעזרת שליחת הודעות טקסט המשתמשים מנחשים את הציור. השחקן המצייר לא יכול לשלוח הודעות טקסט, הדבר מאומת באמצעות הsession ID.

בעת שליחת הודעה ראשית נבדק שהמשתמש השולח לא ניחש כבר תשובה נכונה והוא לא המשתמש שמצייר. אם המשתמש לא ניחש נכון ההודעה תשלח אל יתר המשתמשים.

אם המשתמש ניחש נכון, הניקוד שלו יחושב, הוא יקבל הודעה שהצליח ויבדק אם כל המשתמשים ניחשו. במקרה זה נעבור לתור הבא.

ניקוד למשתמש שלא ניחש בזמן יהיה 0. ניקוד למשתמש שניחש נכון יהיה בטווח שהוגדר מראש וביחס לזמן שלקח למשתמש לנחש מתוך כלל הזמן שהיה.

*תורות*

כל תור נמשך זמן שהוגדר מראש או עד שכל השחקנים ניחשו נכון.

תחילה הניקוד של השחקן המצייר מחושב. הניקוד הוא ממוצע הנקודות של כל שחקן כפול "מס צייר" הנקבע מראש. לאחר מכן, התוצאות החדשות נשלחות לכל שחקן.

לאחר מכן נבדק אם הסיבוב נגמר, אם הוא נגמר בודקים המשחק נגמר ואם כן מראים את לוח התוצאות, אחרת מתחילים סיבוב חדש.

אנחנו זזים מקום אחד קדימה בתור, מודיעים לשחקן המצייר שעליו לצייר ואיזה מילה עליו לצייר ומודיעים ליתר השחקנים שעליהם לנחש את הציור ולבסוף מתחילים מחדש את הטיימר הקובע מתי נגמר התור.

*פרוטוקול התקשורת*

כל הודעה היא בפורמט JSON ויש בה את הערך type הקובע את סוג הפעולה שיש לבצע.

פעולות שנשלחות מהלקוח לשרת

|  |  |  |
| --- | --- | --- |
| Type | פרמטרים | הסבר |
| chatMsg | session, content, username | הודעה חדשה ופענוחה. Session מכיל את הsession של המשתמש לצורכי אבטחה, אימות וזיהוי וcontent מכיל את תוכן ההודעה. Username מכיל את שם המשתמש. |
| join | session | מצרף את השחקן למשחק. Session מכיל את הsession של המשתמש לצורכי אבטחה, אימות וזיהוי |
| mousePress | x, y, session | מקבל עדכון שהצייר לחץ על העכבר ומעדכן את יתר השחקנים על כך. מקבל את שיעורי הנקודה בה לחץ השחקן המצייר (x, y). Session מכיל את הsession של המשתמש לצורכי אבטחה, אימות וזיהוי . |
| mouseReleased | session | מקבל עדכון שהצייר שחרר את העכבר ומעדכן את יתר השחקנים על כך. Session מכיל את הsession של המשתמש לצורכי אבטחה, אימות וזיהוי. |
| colorChange | color, session | מקבל עדכון שהצייר שינה צבע ומעדכן את יתר השחקנים על כך. מקבל את הצבע בפרמטר color. Session מכיל את הsession של המשתמש לצורכי אבטחה, אימות וזיהוי. |
| joinLobby | name, session | מצרף שחקן חדש ללובי. מאמת את השחקן ואת השם שלו בעזרת הname והsession ושומר אותו בזיכרון. |
| startGame | session | מודיע לשחקנים על תחילת המשחק. Session מכיל את הsession של המשתמש לצורכי אבטחה, אימות וזיהוי. |

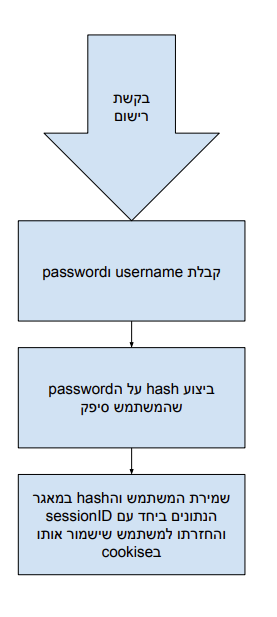
פעולות שנשלחות מהשרת ללקוח

|  |  |  |
| --- | --- | --- |
| Type | פרמטרים | הסבר |
| correctGuess |  | מודיע על כך שהשחקן ניחש נכון. |
| join | content: [chat, canvas] | מודיע על כך שהשרת קיבל את הודעת ההצטרפות ומצרף את היסטורית הקנאס והצ'אט שהלקוח יתעדכן. |
| newPlayer | name\player | הודעה ששחקן חדש הצטרף למשחק\לובי ושמו name\player. |
| chatMsg | content, username | הודעת צ'אט חדשה מעת username שהתוכן שלה הוא content. |
| joinLobby | players | מאשר את הצטרפותו של שחקן ללובי. שולח גם את רשימת השחקנים שנוכחים כעת. |
| startGame |  | מודיע על כך שהמשחק התחיל |
| scoresUpdate | scores | עדכון הניקוד במשחק. scores מכיל את הניקוד החדש. |
| endGame |  | מודיע על סוף המשחק והצגת לוח הניקוד. |
| newRound |  | מודיע על תחילתו של סיבוב חדש. |
| startDrawing | word, time | מודיע על כך שעל השחקן להתחיל לצייר. עליו לצייר את המילה word תוך time שניות. |
| startGuessing | time, drawer | מודיע על כך שעל השחקן להתחיל לנחש את הציור שdrawer מצייר תוך time שניות. |
| joinProhibited |  | מונע מהשחקן להצטרף למשחק בגלל שהוא כבר התחיל |

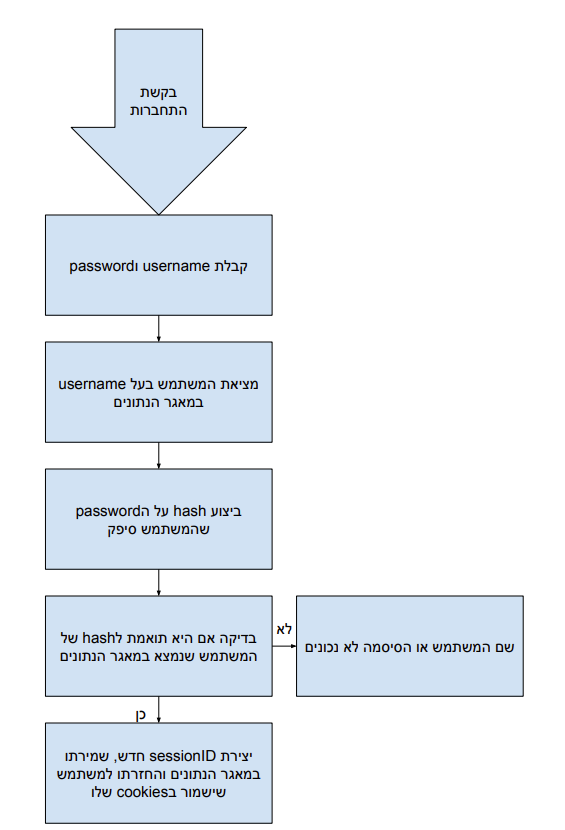
מערכת הauthentication

תפקיד המערכת הוא לאמת את זהות המשתמש

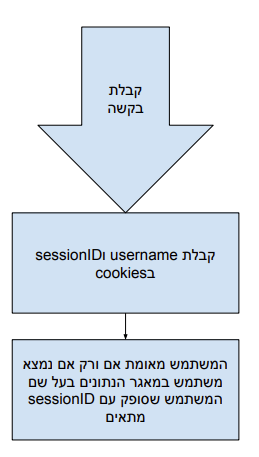
*בקשת רישום*



*בקשת התחברות*



*בקשת אימות*



בטיחות:

* מניעת מתקפות XSS:
  + דוגמה למתקפת XSS שתעבוד ללא יישום אמצעי בטיחות: שלחית ההודעה -

|  |
| --- |
| </div><img src=x onerror=alert(1)//><div> |

* + כדי למנוע זאת, לפני ששם משתמש והודעות מצד משתמש תוצג נקודד תגי HTML מיוחדים (escaping) ונשתמש בספרית DOMPurify.
* אימות משתמש: אנחנו שומרים sessionים של המשתמש ובכך יכולים לאמת את זהותו ולהמנע ממצב של התחזות או רמאות במשחקים. הערה: לכל session אמור להיות גם תאריך בו הוא פג תוקף לצורכי אבטחה ומניעת גניבת sessionים, אך דבר זה לא מומש בפרויקט.
* הצפנת סיסמאות: אנחנו מבצעים hashing לסיסמא, כלומר מצפינים באופן חד כיווני את הסיסמא, ורק אז שומרים אותה במאגר הנתונים. בכך גם אם מאגר הנתונים שלנו מודלף, לתוקפים לא תהיה גישה לסיסמא של המשתמשים.

## מדריך למשתמש

התקנת המשחק:

1. תחילה הורד את המשחק למחשב.
2. עשה activate לvirtual environment.
3. הרץ את הפרויקט בפיתון 3.9
4. פתח את google chrome בhttp://localhost:8080/ (הערה: יש לפתוח כל משתמש נפרד בדפדפן נפרד ולא רק בtab נפרד כדי למנוע חפיפה בין הcookies)

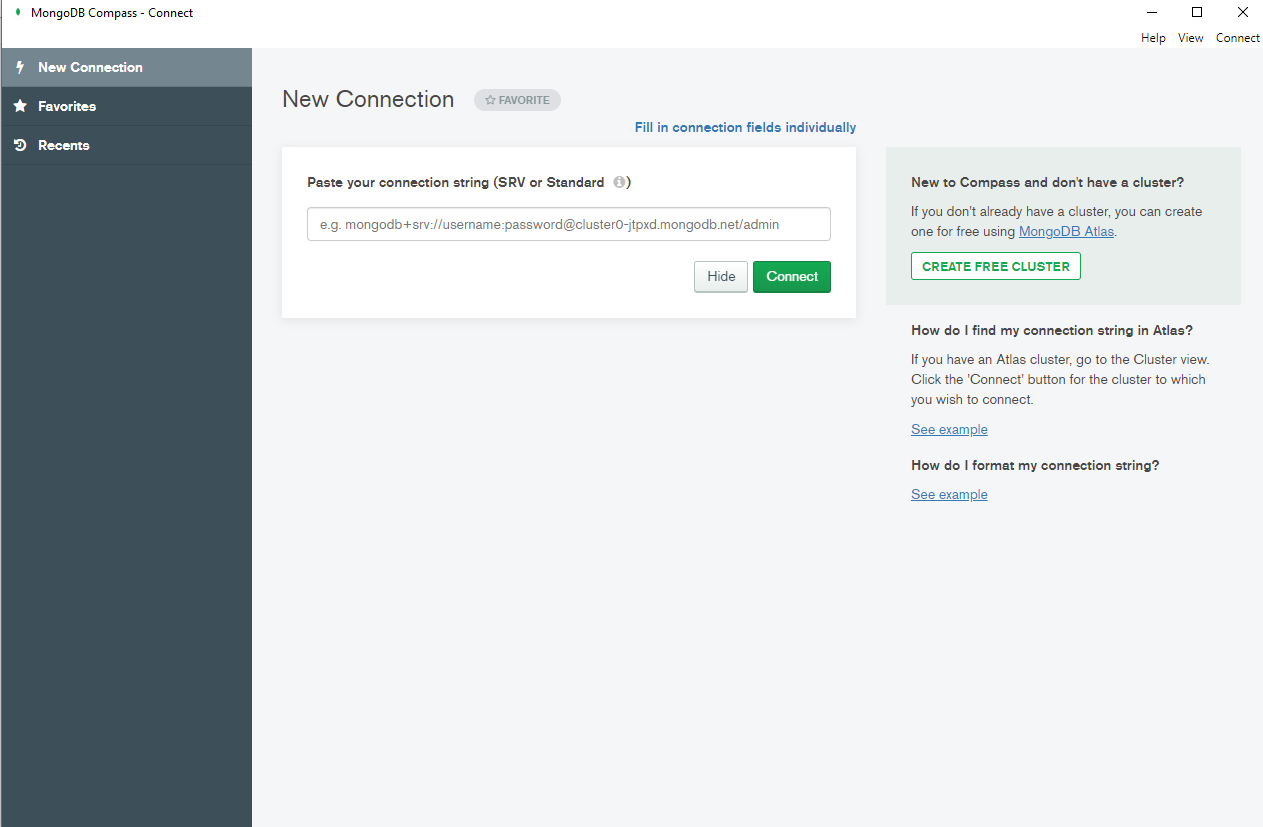
חיבור לבסיס הנתונים

1. קודם כל יש להריץ את בסיס הנתונים באופן מקומי.

בסיס הנתונים הנעשה בו שימוש הוא mongodb.

הורד את שרת בסיס הנתונים MongoDB Community Server בכתובת <https://www.mongodb.com/try/download/community> והתקן.

1. פתח mongoDB Community server



1. השאר את תיבת הטקסט ריקה ולחץ connect. הדבר יריץ שרת באופן לוקלי.