1. ארכיטקטורת המערכת

חלק זה כולל את תיאור מבנה המערכת ופירוט המודולים השונים בה

2.1 מבט על

בסעיף זה יש להסביר את החלוקה העיקרית למודולים השונים במערכת, והיחסים/קשרים ביניהם. יש להגדיר באופן ברור את תחום האחריות של כל מודול או רכיב, ולאיזו פונקציונליות (כפי שהוגדרה במסמך האפיון שנכתב לפני כן) הוא מתקשר (בין אם מודול אחד בלבד או מספר מודולים יחד).  
חשוב להסביר לא רק מה/איך נעשה אלא גם מדוע דווקא כך.

המערכת מחולקת לשלושה מודלים עיקריים.

השרת

הסוכן

האתר

השרת והאתר עובדים בצמידות על אותו מחשב. בעת הרצת השרת, האתר גם הוא עולה בצורה אוטומטית.

הסוכן, הנמצא על מחשבים אחרים (או אותו אחד), שולח הודעות לשרת.

פירקתי את המודלים כך כדי שכל רכיב מערכת יהיה תפקיד עיקרי. השרת אחראי על עיבוד מידע, הסוכן אחראי להעברתו לשרת, והאתר אחראי להציג את התוצאות למשתמש.

2.2 פירוט רכיבי המערכת

**כאן יופיע פירוט לכל רכיב (ניתן לעשות באיטרציות – כל פעם עבור הרכיבים הרלוונטיים, לפני תהליך הפיתוח שלהם)**  
**השרת**

השרת מתחיל את ריצתו על העלתו של האתר כ process נפרד ואז ממשיך בריצתו. (עשיתי זאת כדי שיהיה צורך להריץ רק קובץ אחד – השרת. זה בעיקר לנוחיות. אפשר להריץ את שניהים לכוד והפרוייקט יעבוד אותו הדבר).  
השרת פותח thread שירוץ לאורך שאר ההרצה. תפקידו הוא להאזין לסוכנים חדשים. ברגע שסוכן פונה לשרת, הוא מייצר class שמכיל בשדותיו ערכים המאפיינים את הלקוח ו thread שיטפל בצרכיו וממשיך לחכות לסוכנים נוספים.

thread הסוכן מחכה לקבלת סיכום התעבורה. לאחר שקיבל, מעדכן מספר משתנים לעיבוד עתידי ואז מחכה לסיכום התעבורה הבא. מטרת thread זה הוא פשוט לקלוט מידע ולאגור אותו.

בחזרה ל main – לאחר שפתח את הthread לקבלת סוכנים חדשים, נכנס ללולאה בה בוחר מתי לייצר דוח חדש (על פי הגדרות שקיבל בתחית הריצה כארגומנטים).

כאשר הגיע הזמן ליצור דוח, מדפיס את מספר החבילות הכולל שסוכמו עד כה ומתחיל לייצר את הדוח עצמו ושמירת נתונים חדשים ל DB.

**הסוכן**

הסוכן הינו מאוד נאיבי. מטרתו היחידה היא להסניף תעבורה ולסכם אותה.

הסוכן מתחיל לאגור פקטות (בשימוש של scapy) עד שמגיעה למספר ההוגדר לו (מועבר כארגומנט). כשהגיע לכמות הדרושה, מתחיל לעבור עליהם אחד אחד, ומייצר class בעל שדות מוגדרת מראש המכיל פרטים על הפקטה. לשם מילויי ה class יש פונקציה יחודית שמטרתה למלאות אותו. הסוכן מכין רשימה הבנוייה מהסיכומונים וכשהגיע לסוף, שולח את הרשימה לשרת ומתחיל את התהליך מהתחלה.

**האתר**

האתר הוא החלק היצוגי של הפרוייקט. הוא רשום בשימוש של flask. כאשר משתמש מתחבר לאתר, מוצג בפניו עמוד הבית. שם יכול לבחור להסתכל על דוחות שונים שהשרת ייצר. בנוסף, האתר מייצר סיכום דוחות בשימוש של ה DB שהשרת אגר.

2.3 דיון בנושא העיצוב הנבחר

כאן יש לפרט ובעיקר להסביר מדוע בחרתם דווקא בחלוקת הרכיבים/תפקידים הזו ולא אחרת. מה היתרונות שלכם מבחינתה, וגם מה החסרונות שאתם מודעים אליהם. במידה וחשבתם על חלופות אחרות, יש לציין אותן ולהסביר בקצרה כיצד הן שונות מהפתרון שנבחר - ומדוע החלטתם בסופו של דבר שלא לבחור בחלופות האלו.  
כאן יש להתייחס גם לגבי שפת התכנות שבה בחרתם לכתוב כל רכיב, ומדוע דווקא בשפה זו ולא אחרת.

היתרון בחלוקת הרכיבים לצורה הזו היא שמאוד קל להרחיב ולשנות כל רכיב. ניתן להוסיף שדות נוספים לכל פקטה בקלות, להוסיף עוד עמודים לאתר וניראות הדוחות שהשרת מציג. ניתן גם לשנות את הסוכן השרת והאתר לכל שפה רצויה והפרויקט ימשיך לעבוד כראוי כל עוד צורת המידע (פרוטוקול התקשורת) לא משתנה.

החסרון הוא שלפעמים יש קוד נוסף המכין הרחבה עתידית שלעולם לא ישומש. כלומר כתיבת קוד לא דרוש.

1. עיצוב נתונים ופרוטוקולים

כאן יופיע תיעוד של מבני נתונים שונים / פרוטוקולים המשמשים אותנו במערכת

בסעיף זה יש לפרט את כל סוגי המידע אשר מועבר או נשמר במערכת - בין אם בין רכיבים, בין מודולים או בכל צורה אחרת (כלומר גם אם זה משמש רק לתקשורת פנימית ולא ע"ג הרשת, או למשל אם זה משהו ששומרים לקובץ/בסיס נתונים).

* לכל סוג מידע כזה יש לפרט על השדות אותן הוא מכיל ומאיזה סוג כל שדה, מה טווח הערכים הרלוונטי לגביו וכל הגבלה או מידע נוסף אחר שאתם מוצאים לנכון (למשל: האם מותר שהשדה יהיה ריק ומה המשמעות של זה, אורך או ערך מינימלי/מקסימלי למספר או למחרוזת)
* יש לציין לאיזו מטרה משמש כל מבנה נתונים.
* עבור פרוטוקול תקשורת-נתונים בין רכיבים שונים:
* מה המצבים השונים לכל סוג של תקשורת (למשל אותנטיקציה, הרשמה, שליחת קובץ וכו')
* באילו ייצוגי מידע נעשה שימוש + תרשים זרימה של המצבים השונים שיכולים להיות וסדר השלבים שלהם (למשל כיצד נעביר מידע על מקרים של סיסמה שגויה, משתמש חסום, התחברות מוצלחת וכו')
* מומלץ להגדיר מראש קודים של בקשות/פעולות, תוצאות/תגובות ושגיאות אשר ישותפו בין כך הרכיבים הרלוונטיים
* יש להתייחס לאופי המידע הנשמר (האם הוא טקסטואלי או בינארי) ולייצגו בצורה נוחה והולמת בהתאם
* יש לפרט אודות מבנה בסיס הנתונים (חלוקה לטבלאות, השם והטיפוס של כל עמודה/שדה, קשרים בין טבלאות ואילוצים כלשהם על עמודות)

**פרוטוקול תקשורת**

בפרוייקט יש 2 פרוטוקולים ברשת.  
בין הסוכן והשרת:

הסוכן שולח לשרת רשימה בכל איבר הוא מילון המכיל 6 שדות, כל מילון מייצג סיכום של חבילה. המילון מכיל- שם תוכנה (מחרוזת), כתובת יעד (מחרוזת), מיקום גלובלי/מדינה (מחרוזת), האם החבילה נכנה או יצאה מהמכשיר (בוליאני), פורט בצד הסוכן (מספר) וגודל החבילה (מספר).

בין המשתמש לאתר:

שימוש סטנדרתי של פרוטוקל http- האתר שולח למשתמש את עמוד ה html.

**בסיס נתונים**

בסיס הנתונים בו אני משתמש הינו קובץ סטנדרתי בעל סיומת .dat – הקובץ שומר מחרוזת json של מילון, המכיל כמפתח את שם הדוח וכערך רשימה המכילה – מילון שמכיל כמפתח כתובת מכשיר וכערך את מספר בתים שקיבל, מילון שמכיל כמפתח כתובת מכשיר וכערך את מספר בתים ששלח, מילון שמכיל בכמפתח כתובת מכשיר וכערך התראות (כמו חיבור או התנתקות), מילון המכיל כמפתח מיקום גולובלי/שם מדינה וכערך את גודל המידע שנשלח לשם, מילון המכיל כערך את כתובת היעד וכערך גודל מידע, מילון המכיל כמפתח שם תוכנה וכערך גודל מידע, מילון המכיל כערך מספר פורט וכערך גודל מידע.

4. ממשק משתמש

כאן יופיע פירוט ממשקי המשתמשים וכן תופיע הסקיצה עבורם

בסעיף זה יש לפרט את הפונקציונליות של המערכת כפי שהיא מתבטאת עבור משתמש חיצוני. במידה ויש סוגים שונים של משתמשים, יש להתייחס לכולם בהתאם - אילו רכיבים או נתונים רלוונטיים עבורם, וכיצד הם מתקשרים איתם.  
  
בנוסף, יש לצרף המחשות ויזואליות של המסכים השונים, ולהסביר את התוכן שלהם (למשל מה התפקיד של כל שדה או כפתור, מה קשור/תלוי במה - למשל כפתור שמכובה בהתאם לתנאים מסוימים במערכת וכן הלאה) והקשרים ביניהם (איזה מסך מוביל לאיזה מסך ובאילו מקרים)

ממשק המשתמש בפרויקט שלי הוא למעשה האתר. לאתר יש 4 מסכים עיקריים.

בית

דוח

רשימת כל הדוחות

סיכום

5. נספחים

**כל דבר שהייתם רוצים להוסיף בנוגע לעיצוב המערכת**