**Overseer**

חלק א - מסמך אפיון

אילן דוברומילסקי

0.1v

11/02/2020



**היסטוריית גרסאות המסמך**

|  |  |  |
| --- | --- | --- |
| **תאריך** | גרסה | **תקציר השינויים** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**תוכן עניינים**

**1. הקדמה**

החלק הזה ישמש להצגה כללית של מסמך האפיון - הוא נועד לאפשר לקורא להבין במה מדובר - מה זה בכלל הפרויקט הזה ומה מופיע בהמשך המסמך.

1.1 מטרה

*מה המטרה של המסמך ומי קהל היעד שלו*

מטרת המסמך הינו הצגה כללית של הפרויקט, כיצד הוא עובד ולמה הוא מיועד.

1.2 המוצר

*חלק זה צריך להכיל את הדברים הבאים:*

* *שם המוצר אותו מפתחים (ובאמצעותו נתייחס למוצר בהמשך המסמך)*
* *הסבר כללי של מה המוצר עושה (ואם צריך- מה הוא לא עושה)*
* *תיאור של מטרות המוצר ויתרונותיו העיקריים. נסו להיות מדויקים וספציפיים כמה שניתן.*

*שם מוצר: Overseer*

*מה המוצר עושה: המוצר מייצר דוחות המסכמים פרטים על תעבורת הרשת בפרקי זמן או כמות המוגדרים על ידי המשתמש. בנוסף, יכול להציג סיכום של כל הדוחות.*

יתרונות: דוחות המוצר מוצגים בצורה קריא המאפשר ניתוח והבנה פשוטה ומהירה. המוצר קל לשימוש. המוצר הוא בעל רמת דיוק גבוהה.

1.3 הגדרות

*בחלק זה יש לציין את כל ההגדרות, המושגים וראשי התיבות בהם תשתמשו בהמשך המסמך.*

שרת: השרת הוא ה"מוח" של המערכת. הוא דואג לניתוח ויצירת הדוחות. לצידו פועל האתר.

אתר: מציג את הדוחות הקיימים למשתמש.

סוכן: נמצא על מחשב העובדים. אחראי על קליטת וסיכום התעבורה ושליחתו לשרת.

פקטה: יחידת מידע העוברת ברשת.

DB(database) – קובץ המכיל json של מידע דרוש לפעולות בין הרצות.

**1.4** תקציר

*הסבר מה מופיע בהמשך המסמך ופירוט כיצד זה מסודר לאורך המסמך*

1. תיאור כללי

החלק הזה מתאר את הגורמים העיקריים המשפיעים על המוצר ועל דרישותיו. חלק הזה לא מפרט דרישות ספציפיות אלא רק עוזר להבין את הדרישות.

2.1 פונקציונליות

*חלק זה יכיל סיכום של הפונקציות העיקריות במערכת (לא פונקציות של שפת תכנות, אלא באופן מילולי - מה הדברים שהמערכת עושה)*

שרת:

* מריץ את האתר.
* מנתח את החבילות שנשלחות אליו מכל הסוכנים.
* מייצר דוח.

אתר:

* מציג את הדוחות.
* מייצר דוח - סיכום דוחות.

סוכן:

* מסניף פקטות היוצאות ונכנסות למחשב.
* עושה סיכום של כל פקטה על פי שדות קבועות מראש.
* שולח לשרת את לניתוח כללי את הפקטה במסוכמת.

2.2 קהל היעד

*תיאור כללי של קהל היעד של המערכת - מה תפקידם, ניסיונם, יכולתם ללמוד את ממשק המערכת וכל דבר אחר שישפיע על דרישות המערכת. יתכן יותר מפרופיל אחד של משתמש במערכת, ולכן יש לפרט את כל הסוגים וההשלכות/משמעויות שלהם מבחינת אופי ורקע טכנולוגי או פונקציונלי.* ("מי הולך להשתמש במערכת?")

המוצר מיועד לכל ארגון המעוניין לפקח/להשיג מידע על תעוברת הרשת וכל עובד בפרט.

דרוש מעט ידע על רשתות. המשתמש מריץ את השרת – ביכולתו לבחור פורט אליו הסוכנים יפנו, כל כמה זמן ייצר דוח ובאיזה אופן. סוכן – פורט אליו יפנו הסוכנים בשרת, כתובת השרת.

על פי הגדרות אלו, המשתמש צריך לדעת למצוא את כתובת אינטרנט של השרת, ולדעת באיזה פורט לבחור כך שלא יפגע בתוכנות אחרות.

2.3 אילוצים עיקריים

*הגבלות ואילוצים עיקריים שצריך לקחת בחשבון בזמן עיצוב המערכת - אילוצי חומרה, סביבת ריצה, זמן ריצה וכו'..* (למשל – "המערכת דורשת חיבור לאינטרנט" / "המערכת צריכה לתמוך במספר רב של מערכות הפעלה" / "המערכת צריכה לתמוך במספר משתמשים רב")

בשביל שהמוצר יעבוד כמו שצריך, דרושים לפחות שתי מחשבים – אחד שיריץ את השרת, ואחד שיריץ את הסוכן.

על המחשבים דרושה השפה פייתון גרסת 3.7.4 לפחות עם הספריות os, socket, json, time, threading, subprocess, sys, requests, re, scapy, flask. לשרת צריך להיות מספר פנוי של פורטים ככמות הסוכנים +1.

2.4 הנחות ותלויות

*כל הנחה שיש לנו לגבי המערכת וסביבת הריצה שלה, ששינוי שלה עשוי לגרום לשינוי באפיון ובעיצוב - לדוגמא "המערכת תרוץ בסביבת לינוקס" זו הנחה המשפיעה על אפיון ועיצוב המערכת. אם נשנה את זה בהמשך נצטרך לעבור שוב על המסמך ולשנות את אפיון ועיצוב המערכת בהתאם (זה שונה מהסעיף הקודם- אילוצים זה לא משהו שיכול להשתנות, זה משהו שנקבע בתחילת העבודה ומשפיע רק על האפיון הראשוני)*. אל תתנו להנחות הללו להגביל אתכם בהמשך (אם תגלו שלמשל עדיף בסוף לרוץ מעל וינדוס – עשו זאת).

המוצר עובד בעיקרו מעל הרשת. הוא נרשם ב windows אך אני מאמין שיכול לעבוד מעל כל המערכת הפעלה התומכת באילוצים.

**2.5** סיקור מצב השוק כיום (אופציונלי)

*במידה וידוע לכם על מוצרים קיימים כיום (אתרים, תוכנות, אפליקציות וכו') שנותנים פתרון דומה (גם אם חלקי) לרעיון שלכם - קשרו אליהם כאן או תארו אותם בקצרה. התמקדו באופן שבו הם פותרים או מתייחסים לסוגיות דומות לאלו שבחרתם, מה הפיצ'רים העיקריים שהם מציעים ובמה הם שונים מהמוצר שלכם.*

https://www.solarwinds.com/netflow-traffic-analyzer/use-cases/network-traffic-analysis

wireshark

1. דרישות מפורטות

החלק העיקרי והחשוב ביותר של מסמך האפיון. פרק זה יכיל את הדרישות המפורטות מהמערכת שינחו את מעצבי המערכת, המפתחים, והבודקים בהמשך. כל דרישה שתפורט בחלק זה צריכה להיות:

* מזוהה על ידי שם (או עדיף - מספר) באופן ייחודי
* מתועדפת (כלומר מה העדיפות שלה כלפי הדרישות האחרות)
* ניתנת לבדיקה
* נכונה ושלמה
* עקבית ולא מתנגשת עם דרישות אחרות

*שימו לב שישנם מספר סוגים של דרישות, כפי שמפורט בכותרות כאן:*

*יש להשקיע בחלק זה ולפרט את הדרישות בצורה ברורה ונוחה להבנה. יש לזכור שזה לא מסמך עיצוב- יש להמנע מתיאור מפורט מדי שמגביל את העיצוב הטכני (למשל – שפת תכנות) שיגיע בשלב מאוחר יותר של עבודתכם.*

*את הדרישות רצוי לחלק לקטגוריות הבאות:*

3.1 דרישות פונקציונליות

*תיאור של פיצ'רים ספציפיים במערכת. לכל דרישה פונקציונלית מומלץ לפרט:*

* *הסבר כללי*
* *קלט*
* *תהליך*
* *פלט*
* *טיפול בשגיאות*

*ניתן (ואף רצוי) להציג את הדרישות הפונקציונליות בפורמט של [תרחיש](http://he.wikipedia.org/wiki/%D7%AA%D7%A8%D7%97%D7%99%D7%A9_%D7%A9%D7%99%D7%9E%D7%95%D7%A9)* [*שימוש (use case)*](http://he.wikipedia.org/wiki/%D7%AA%D7%A8%D7%97%D7%99%D7%A9_%D7%A9%D7%99%D7%9E%D7%95%D7%A9)*. בכל "נקודת זמן" של תיאור התרחיש יש לפרט את כל האפשרויות השונות שעומדות בפני המשתמש או המערכת, ומה יכול לקרות בכל התפצלות שכזו (שימו לב להתייחס גם למקרים שליליים, כמו שגיאות או נתונים לא מתאימים, ולא להניח שהמשתמש תמיד פועל בדרך שלה אנו מצפים)*

3.2 דרישות של ממשקים חיצוניים

**3.2.1 ממשקי משתמש** (למשל: GUI)

האתר משמש כממשק משתמש.

**3.2.2 ממשקי חומרה**

**3.2.3 ממשקי תוכנה** (אם התכנה מחולקת לכמה תוכנות שונות שצריכות לתקשר בניהן)

**3.2.4 ממשקי תקשורת (למשל תקשורת בין שר**ת ללקוחות)

3.3 דרישות לא פונקציונליות

*דרישות מהמערכת שלא מתבטאות בפיצ'ר ספציפי או בתהליך ספציפי שמתרחש במערכת אבל משפיעות על אופן עיצובה ומימושה, לדוגמא:*

**3.3.1 דרישות ביצועים (performance)**

קוד השרת בנוי ברובו מ threading. החילוק משפר ביצועים

**3.3.2 דרישות מהימנות (reliability)**

**לאורך הקוד יש הקפדה על שמירת נתונים בזמנים אופטמאלים,דאגה על כך שהקוד לא יקרוס משגיאות.**

**3.3.3 דרישות זמינות (Availability)**

**3.3.4 דרישות אבטחה (security)**

**3.3.5 דרישות תחזוקה (maintainability)**

**3.3.6 דרישות ניידות (portability)**

3.4 דרישות בסיס נתונים

*הנתונים הנכנסים לאחסון הם כל השדות הדרושות ליצור דוח. כך אם יש צורך בשחזור דוח, אפשר להעיר את הנתונים לפונקציה בשרת והוא ידע ליצר את הדוח בשנית.*

*הנתונים נשמרים כ json בקובץ מסוג .dat. אין חשיבות לסוג הקובץ, אלה רק למבנה התוכן שחשוב שהיה json שפייתון יכול לקרוא.*

*נפח האחסון הדרוש הוא תלוי זמן ריצה. ככל שהשרת רץ יותר זמן, כך ייצר יותר דוחות שישמרו ב DB. אומנם משקלו של הנתונים של כל דוח אינו הרבה, עם הזמן יגדל.*

*הקובץ עצמו אינו מוצפן בשום צורה ואף קריא יחסית.*

3.5 דרישות נוספות

*כאלו שלא ידעתם לאיזה חלק הם שייכים*

1. כלי ניתוח נוספים

*בחלק זה אפשר להוסיף דיאגרמות שונות שיעזרו לנתח ולהבין את התנהגות המערכת (יכול להיות שימושי במיוחד חלק של פירוט תרחישים ומצבים שונים בפונקציונליות של המערכת).*

*לדוגמא (מוזמנים לחפש על הנושא):*

**4.1 Sequence Diagrams**

**4.2 Data Flow Diagrams (DFD)**

**4.3 state-Transition Diagrams (STD)**

5. נספחים