**<TRIVR>**

מסמך אפיון

<יותם לויט>

<גרסה X>

<תאריך>

**היסטוריית גרסאות המסמך**

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



1. **הקדמה**

1.1 מטרה

*מטרתו של מסמך זה, היא להציג את הפרויקט בצורה מפושטת ואת חלקיו. המסמך הזה מיועד לכל מי שרוצה להבין לעומק את הפרויקט ואיך הוא עובד בצורה פשוטה ככל שניתן בלי להיכנס לקוד.*

* 1. המוצר

*שם המוצר הוא* TRIVER – Test Robot In Virtual Reality . TRIVER *הוא סימולטור בדיקה אשר מטרתו להקנות סביבה וירטואלית, שבה ניתן לבחון רובוטים ללא חשש לפגיעה פיזית בהם או בסביבה*. *יתרון TRIVER על פני בדיקה של הרובוט בעולם הפיזי היא שעל ידי בדיקה בסימולטור בעלי הרובוט יוכלו להימנע מתאונה מסוכנת בעולם הפיזי.*

* 1. הגדרות

**לסמלץ – להפעיל סימולטור**

**לקוחclient – משתמש המסמלץ את הרובוט שלו בסימולטור**

**שרת – מחשב מרוחק בו TRIVER מורץ** .

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

1. בדיקה ווירטואלית

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

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?")

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")

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 להריץ את השרת המרכזי מעל וינדוס – עשו זאת).

**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.*

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 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)

**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)**

**3.3.2 דרישות מהימנות (reliability)**

**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

* *איזה נתונים יישמרו*
* *באיזה פורמט יש לשמור את המידע (DB / קובץ . ניתן לציין גם כמה אופציות)*
* *מה יכולות האחסון הנדרשות (נפח אחסון לדוגמא)*
* *מה רמת האבטחה הנדרשת*

3.5 דרישות נוספות

*כאלה שלא ידעתם לאיזה חלק הם שייכות*

5. נספחים