**<Cryptocurrency>**

מסמך אפיון

<איילון נוה>

<גרסה 1>

<תאריך>

**היסטוריית גרסאות המסמך**

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



**1. הקדמה**

1.1 מטרה

*המטרה של המסמך היא להסביר את דרישות הפרויקט- Cryptocurrency*

1.2 המוצר

* *שם המוצר הוא cryptocurrency*
* *המוצר הוא המטבע הדיגיטלי והטכנולוגיה מאחוריו. בנוסף יש מערכת פיקטיבית שלו ושל מטבעות אחרים ושווין ובוטים הסוחרים בהם. יתר על כן, תהיה מערכת המנתחת את שוויו.*
* *מטרת המוצר היא בעצם מטבע סחיר (כמו דולר) שיהיה דיגיטלי (כמו ביטקוין). המטבע יהיה עם הבטחה גבוהה בעזרת טכנולוגית בלוקצ'יין.*

1.3 הגדרות

*מטבע דיגיטלי ( (Cryptocurrency- המטבע הדיגיטלי מונפק על ידי גוף פרטי כלשהו שמפתח טכנולוגיה שאינה מאפשרת לשכפל אותו. המטבע בדרך כלל אינו שייך לגוף שהנפיק אותו אלא פתוח לכולם. למעשה מדובר בפרוטוקול (תוכנה), שרץ במקביל על אלפי מחשבים ונרשם על ידיהם.*

*בלוק צ'יין - היא תפיסה טכנולוגית במחשוב המאפשרת פעילות עסקית מאובטחת באינטרנט ואימות של טרנזקציות עסקיות בין צדדים שונים ללא צורך בישות ניהול מרכזית.*

*ביטקוין - הביטקוין הוא מטבע שאינו תלוי בגוף מרכזי. במקום זאת, ביטקוין הוא רשומה בקובץ ציבורי המכונה "שרשרת בלוקים" (בלוקצ'יין), ניתן לנהל אותו או להשתמש בו בעזרת תוכנה המכונה "ארנק". מערכת הביטקוין היא מערכת המבטיחה שהתשלומים יתבצעו באופן תקין כך שסכום שהועבר בתשלום אכן נגרע מארנק המקור והתווסף לארנק היעד.*

*כרייה – הוספת בלוק לשרשרת הבלוקים על ידי תהליך מסובך הכולל חישובים מתמטיים ודורש יכולות מחשוב גדולות*

**1.4** תקציר

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

1. תיאור כללי

2.1 פונקציונליות

*המערכת מחולקת לכמה חלקים שיצרו את הבלוק צ'יין, את הארנק הכולל ממשק גרפי ואת הסחר במטבע.*

2.2 קהל היעד

*קהל היעד של המערכת הוא בעצם כל אדם שרוצה לרכוש, לסחור ולהעביר את המטבע הווירטואלי שאני יוצר.*

*המשתמשים במערכת צריכים הבנה בסיסית במטבעות, כלומר, הם צריכים לדעת שערכם משתנה ומה היתרונות והחסרונות של להחזיק מטבע ווירטואלי. המערכת אינה מתאימה לילדים.*

2.3 אילוצים עיקריים

**- המערכת דורשת חיבור לאינטרנט**

**- המערכת צריכה לתמוך במספר רב של משתמשים**

**- השרת שיוצר את הבלוקים ואת הכרייה יצטרך יכולות מחשוב גבוהות**

**- המערכת צריכה אבטחה גבוהה מאוד**

2.4 הנחות ותלויות

**- המערכת תרוץ על ווינדוס**

**- המטבע לא יסחר גם כמטבע אמתי וגם במערכת עצמה**

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

ישנם כמה מטבעות העובדים בטכנולוגיה כמו שאני ממש. הגדולים בהם הם: ביטקוין ואת'ריום שלשניהם שווי שוק של עשרות מיליוני דולרים.

בנוסף טכנולוגית בלוק צ'יין משמשת כיום ליצירת חוזים ולמימושם באופן אלקטרוני, העברות רשומות רפואיות באופן מאובטח, ועוד.

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)*

*פונקציות בארנק*

*פונקציות הנתונות לשימוש על ידי המשתמש במערכת דרך הארנק (הממשק משתמש שלו)*

קניית מטבעות

קניית המטבע הווירטואלי על ידי תשלום במטבעות אחרים

קלט: פרטי האשראי וכמות הכסף שרוצים לשלם, על מנת לקבל את המטבעות

תהליך: שליחת הבקשה לשרת עם הפרטים ועדכון הארנק של המשתמש בהתאם

טיפול בשגיאות: בדיקה עם השרת שפרטי האשראי נכונים

העברת מטבעות

העברת מטבעות למשתמש אחר

קלט: כמות המטבעות שהמשתמש רוצה להעביר ואת פרטי המשתמש שהוא רוצה להעביר אליו

תהליך: שליחת הבקשה ואת פרטיה לשרת

טיפול בשגיאות: בדיקה שלא מעבירים כסף שאין בארנק, ובדיקה שהכתובת שלי מועברים היא חוקית

מכירת מטבעות

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