**Draw Something**

מסמך אפיון

<קרני בר >

<גרסה 0.0>

<12/12/2018>

**היסטוריית גרסאות המסמך**

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



**מסמך אפיון**

**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.

המטרה:

מטרת מסמך האפיון הוא ליצור סדר בתכנון של בניית הפרויקט. המסמך יאפשר לי להיעזר בו כבסיס שינחה אותי למטרות הבסיסיות שלי לפרויקט.

קהל היעד של המסמך הוא אליי (יוצרת הפרויקט) ואליכם המורים.

המוצר:

הפרויקט שלי הוא מין צ'אט קבוצתי הנקרא draw something ובו החברים בקבוצה שולחים אחד לשני ציורים והראשון בקבוצה שמנחש נכון את מה שציירו זוכה בנקודות. החברים שבקבוצה מתחרים על מי יש יותר נקודות כך שכל השחקנים שואפים לצבור כמה שיותר נקודות במשחק.

**המשחק הוא השראה של משחק קיים הנקרא draw something**

קישור למסמכים קודמים:

קישור למצגת-https://docs.google.com/presentation/d/1zwwhFIVrwksSeU6UzpR9BHxZtQ0ZnE22hfAd6DjSHKM/edit?usp=sharing

הגדרות:

GUI- Graphic User Interface

OOP- Object Oriented Programming

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

*המערכת תקבל מהמשתמש הודעות מילוליות או את הציור שלו וותצטרך להעביר ולשלוח את ההודעה הזאת לשאר חברי הקבוצה. המערכת תתמוך בשימוש של רב משתמשים בו זמנית באמצעות threading ושליחת והעברה ותיאום מידע בין השחקנים באמצעות שליחת פקטות tcp.*

2.2 קהל היעד

**המשחק נועד לכל אחד, כל מי שרוצה בידור\משחק קליל להעברת הזמן.**

2.3 אילוצים עיקריים

-המערכת צריכה לתמוך במספר משתמשים בו זמנית

-האפליקציה צריכה להתאים לכל מכשירי האנדרואיד

-האפליקציה דורשת חיבור לאותה רשת אינטרנט

-שליחת הציורים עלולה להעמיס על המערכת

2.4 הנחות ותלויות

**המערכת תפעל על מכשירי אנדרואיד**

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

**ממשקי משתמש-** GUI

**ממשקי חומרה- סמארטפון אנדרואיד**

**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)**

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