***DevStegram***

**מסמך אפיון**

**איתי קבנובסקי**

*<גרסה 0.00>*

*<25.10.2021>*

**היסטוריית גרסאות המסמך**

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



**1. הקדמה**

הפרויקט ישמש כרשת חברתית למפתחי תוכנה. כגון אינסגרם טוויטר וכדומה.

# 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

*מטרת המוצר היא לספק למתכנתים מכל רחבי העולם רשת חברתית הנותנת אפשרות לתקשר עם חברת המתכנתים ולהעלות פוסטים. היא גם תתאים את עצמה לתחומי העניין שלהם.*

1.2 המוצר

* *שם - DevStegram*
* *מטרה - נותן אפשרות למתכנתים לתקשר אחד עם השני דרך פלטפורמה אינטראקאיבי שמתאימה את עצמה למשתמש.*

*- לומד את תחומי העיניין של המשתמש וממליץ לו פוסטים דומים להתעניינותו ברחבי הפלטפורמה.  
המוצר מספק אפשרות עוקב ונעקב וכך יקח בחשבון גם אותם בחישוב תחומי העיניין.  
סה"כ זה בסגנון רשתות חברתיות המוכרות לנו כיום*

1.3 הגדרות

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

*בינתיים יש רק אחד וזה פרוס-process, כל תוכנה שפועלת בזמן אמת.*

**1.4** תקציר

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

1. תיאור כללי

2.1 פונקציונליות

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

* *לוגין ורגיסטריישן, מבקש תחומי עיניין*
* *מחשב מודל הצעות בהתאם לפוסטים הראשנים שנבחרו*
* *מציע פוסטים לפי מודל הצעות שחושב*
* *נותן אפשרות לחיפוש פוסטים, משתמשים*
* *נותן אפשרות לעקוב אחרי משתמשים (משפיע על חישוב המודל)*
* *בסוף מחזור מחשב מודל הצעות מחדש, זמן מחזור יקבע בהמשך (כנראה פעם ביום)*
* *חוזר חלילה*

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?")

כל מתכנת מתחיל ממוצע ומתקדם יוכל להשתמש בפלטפורה כדי לתקשר ולהתעניין. הם יכולו לקבל מידע מותאם בשבילם ובכך ליעל את תהליך החיפוש שלהם.

2.3 אילוצים עיקריים

יעילות שמירה וקריאה מדאטא בייס – קלאוד בייסד דאטא בייס ופויינטרים לתמונות , מה קורה אם אין למשתמש חיבור אינטרנטי – לשמור רקווסט בתור, המערכת צריכה לתמוך בהרבה משתמשים.

2.4 הנחות ותלויות

אין בינתיים משהו כזה

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

אינסטגרם, טוויטר, פייסבוק, טאמבלר ... – רשתות חברתיות

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- נתוני הממוצע של כל פרוס*
* *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