Cloud server

מסמך אפיון

דניאל וולקוביץ

ת.ז: 207257668

גרסה 2.0

8.6.2016

**היסטוריית גרסאות המסמך**

|  |  |  |
| --- | --- | --- |
| **תאריך** | גרסה | **תקציר השינויים** |
| 2.3.2016 | 1.0 | בניית מסמך האפיון לקראת תחילת הפרויקט |
| 15.3.2016 | 1.1 | שינויים ב use cases |
| 8.6.2016 | 2.0 | שינויים לקראת הגשת הפרויקט |



**1. הקדמה**

1.1 מטרה

מטרת המסמך היא להציג את מערכת Cloud Server רעיונית, את הדרישות עם ההיבט התפעולי שלהן ואת ניתוח הדרישות . מסמך זה משמש כבסיס למסמך תפעול של המערכת עבור משתמשים, וגם משמש את בסיס לתכנון, עיצוב ופיתוח המערכת. בנוסף המסמך ישמש לבדיקת המערכת על ידי יצירת תסריטי בדיקה.

1.2 המוצר

**מערכת Cloud Server, מכאן ואילך יקרא הפרויקט בשם: "ענן".**

**ה"ענן" מאפשר לכל משתמש רשום, ז.א. שנרשם בעזרת התוכנה ונמצא במאגרי השרת, לסנכרן את קבציו הנמצאים על מקום אחסון פרטי כגון קונן קשיח, על גבי ה"ענן".**

**המשתמש יכול להסיר דאגה מליבו בידיעה שהמידע שלו נמצא מגובה ב"ענן". אם תקרה למחשב שלו תקלה, הקבצים עדיין יהיו קיימים ונגישים ב"ענן".**

**המשתמש מתקין על המחשב שלו אפליקצית לקוח ודרך ממשק משתמש הוא בוחר תיקייה (או מספר תיקיות) שיסונכרנו אל ה"ענן". כמוכן יוכל המשתמש לשחזר ולהוריד קבצים שהעלה בעבר ל"ענן".**

**המטרה היא לספק למשתמש ממשק פשוט וטבעי שמתחבר עם צורת השמירה הקיימת, יכולת לסנכרן את קבציו ותיקיותיו ל"ענן". תמיד ניתן לאחזר, במקרה הצורך, את הקבצים מה"ענן".**

1.3 הגדרות

* *ענן: בפרויקט זה משמעות המילה "ענן" הינו שרת המסנכרן קבצים ממספר בלתי מוגבל של לקוחות ולאו דווקא מאפשר שיתוף של הקבצים הנ"ל.*
* *תיקיית סנכרון ראשית: אצל הלקוח קיימת תיקייה ממנה הוא מסנכרן את קבציו ותיקיותיו. תחת תיקיית הסנכרון הזו מותר למשתמש להניח תיקיות בלבד (כמובן שמתחתיהן מותר להניח גם קבצים) כל תיקייה כזו תיקרא תיקיית סנכרון ראשית. התיקייה נמצאת באותו ה-*path *שממנו רצה תוכנית הלקוח ששמה הוא:* synchronizationfolder.

1.4 מבנה המסמך

*המסמך בנוי מהפרקים הבאים:*

*פרק 1 הקדמה:*

*הסבר על מטרת המסמך, מטרת המערכת והגדרות כלליות.*

*פרק 2 תיאור כללי:*

*תיאור כללי של כיצד יכולות המערכת ולמה היא מיועדת.*

*פרק 3 דרישות מפורטות:*

* + - דרישות פונקציונליות: מפרטות את ההתנהגות של המערכת. הדרישות נאספו כ-use cases על מנת להסביר את ההיבט התפעולי ואיך המערכת מגיבה לכל בקשת משתמש.
    - דרישות ממשקים: מפרטות את כל הממשקים עם העולם החיצון.
    - דרישות לא פונקציונליות: מפרטות את כל האילוצים בהם המערכת צריכה להתחשב או לעמוד בהם.

1. תיאור כללי

2.1 פונקציונליות

**הרשמה למערכת: המערכת תאפשר לכל משתמש חדש להירשם לתוך מאגר המשתמשים הרשומים שלה.**

**התחברות והתנתקות: המערכת תאפשר לכל משתמש קיים להתחבר אל המערכת ולהתנתק ממנה.**

**בחירת תיקיות לסינכרון: המערכת תמקם ב-**path **שממנו היא רצה תיקייה, כל תכולת התיקייה תסונכרן אל השרת.**

**סינכרון: המערכת תסנכרן באופן אוטומטי את התיקייה כל עוד המשתמש מחובר אל המערכת ותוכנת הלקוח רצה.**

**סינכרון מהשרת: ברגע חיבור המשתמש למערכת יסונכרנו לתיקיית הסנכרון שלו כך הקבצים שסונכרנו בעבר אל הענן.**

**כיווץ: בעת תעבורת הקבצים בין השרת ללקוח המערכת דואגת לכווץ את הקבצים על מנת לאפשר תעבורה מהירה ויעילה.**

**הצפנה: סיסמאות המשתמשים יישמרו ע"י** HASH **שיחושב להם על מנת לאבטח את הסיסמאות.**

2.2 קהל היעד

**קהל היעד הוא כל משתמש אשר מעוניין לסנכרן את קבציו אל הענן. בכך יוכל המשתמש לשמור לעצמו גיבוי של קבצים חשובים שיהיו זמינים אליו בכל עת ובכל מקום. המערכת פשוטה למדי ואין צורך בניסיון מקדים בתכנות על מנת להשתמש בה. המשתמש זקוק להיכרות בסיסית עם נושא ה"ענן" בלבד ולא ליתרה מזו. על כן נעשה ניסיון לשמור על המערכת פשוטה ונוחה לשימוש על מנת לאפשר חוויית משתמש מקסימלית.**

2.3 אילוצים עיקריים

* **המערכת דורשת חיבור לאינטרנט.**
* **המערכת תומכת במספר רב של משתמשים במקביל.**

2.4 הנחות ותלויות

**המערכת תרוץ בסביבת ווינדוס.**

2.5 סיקור מצב השוק כיום

**כיום קיימים שירותי ענן רבים כאשר המוכרים שבהם הם:** dropbox, Google drive, icloud. **השימוש בענן הולך וגובר, ישנם אפילו עסקים שמסתמכים על עמדות עבודה שמסונכרנות לענן באופן מלא כך שלמשתמש קצה אין מחשב יחיד שממנו הוא עובד.**

1. דרישות מפורטות

3.1 דרישות פונקציונליות

**Use Case הרשמה למערכת:**

**תיאור: המשתמש מתחבר אל המערכת בפעם הראשונה. המערכת תציג בפני המשתמש אפשרות להיכנס או להירשם. במקרה זה הלקוח בחר להירשם. המשתמש יתבקש בעזרת** GUI **להזין שם משתמש וסיסמא. מעכשיו בעזרת פרטים אלו יוכל הלקוח להיכנס אל חשבונו מכל מחשב. במקרה ששם המשתמש כבר קיים, תבקש המערכת להזין שם משתמש אחר.**

**תנאי קדם: אין**

**תנאי סופי: המשתמש רשום ושמור במערכת עם שם משתמש ייחודי משלו ועם סיסמה.**

**זרימה רגילה:**

1. **המשתמש מתחבר למערכת ובוחר להירשם במערכת**
2. **המשתמש מזין שם משתמש.**
3. **המשתמש מזים סיסמא.**
4. **המערכת בודקת אם השם כבר קיים. אם הוא קיים היא מחזירה הודעה שהשם קיים במערכת.**
5. **המערכת בודקת תקינות הסיסמה. אם הסיסמה לא תקינה היא מחזירה הודעה בהתאם.**
6. **המערכת שומרת את שם המשתמש עם ה-**HASH **שחושב לסיסמא.**
7. **לאחר שהמשתמש ייגש לחשבונו בהצלחה המערכת תחל בניתור על כל השינויים הנעשים בתיקיית הסנכרון.**

**חריגים: אין.**

**אלטרנטיבות: כניסה לחשבון משתמש.**

**Use Case כניסה לחשבון משתמש:**

**תיאור: הלקוח מתחבר לחשבון משתמש שלו על מנת לקבל אפשרויות הפעלת המערכת. המשתמש מקבל GUI על מנת להזין שם משתמש וסיסמה. רק משתמש וסיסמא כפי שהם רשומים במערכת מאפשרים למשתמש לקבל תפריט פונקציות הפעלה.**

**תנאי קדם: הלקוח כבר רשום במאגרי השרת.**

**תנאי סופי: הלקוח יקבל גישה אל חשבון המשתמש הפרטי שלו.**

**זרימה רגילה:**

1. **המערכת תציע לבצע כניסה לחשבון והמשתמש יבחר באופציה זו.**
2. **המערכת בודקת אם השם כבר קיים. אם הוא לא קיים היא מחזירה הודעה שקיים פרט שגוי בהזנת הפרטים.**
3. **המערכת בודקת תקינות הסיסמה. אם הסיסמה לא תקינה היא מחזירה הודעה שיש פרט שגוי.**
4. **לאחר שהמשתמש ייגש לחשבונו בהצלחה המערכת תסנכרן אל המשתמש את הקבצים שהעלה בעבר לשרת ואחרי כן המערכת תחל בניטור על כל השינויים הנעשים בתיקיית סנכרון.**

**חריגים: אין.**

**אלטרנטיבות: הרשמה למערכת.**

**Use Case יציאה מחשבון משתמש:**

**תיאור: הלקוח מחובר למערכת והוא רוצה להתנתק מהשירות.**

**תנאי קדם: הלקוח כבר מחובר אל חשבון המשתמש שלו.**

**תנאי סופי: הלקוח יסומן כמנותק מחשבונו. הוא לא יוכל להשתמש בשירות עד החיבור הבא.**

**זרימה רגילה:**

1. **המשתמש בוחר להתנתק מהמערכת וסוגר את התוכנה.**
2. **המערכת תבצע יציאה מן החשבון.**
3. **המערכת מסמנת שהמשתמש מנותק מהחשבון.**

**חריגים: אין.**

**אלטרנטיבות: אין.**

**Use Case מחיקה:**

**תיאור: המשתמש מוחק קובץ\תיקייה מתיקיית הסנכרון.**

**תנאי קדם: הלקוח כבר מחובר אל חשבון המשתמש שלו.**

**תנאי סופי: סנכרון השרת בדגש על מחיקת קבצים.**

**זרימה רגילה:**

1. **הלקוח ימחק את הקובץ\תיקייה מהתיקייה המסונכרנת שלו במחשב.**
2. **השרת ימחק אותם גם אצלו , אם מדובר בתיקיית סנכרון ראשית אזי הוא גם יימחק מרשימת התיקיות.**

**חריגים: אין.**

**אלטרנטיבות: אין.**

**Use Case יצירה:**

**תיאור: המשתמש יוצר קובץ\תיקייה חדש\ה בתיקיית הסנכרון.**

**תנאי קדם: השרת כבר סנכרן אל המשתמש את תיקיות הסנכרון הראשיות.**

**תנאי סופי: השרת יצור בזיכרון שלו את אותו קובץ\תיקייה שיצר המשתמש.**

**זרימה רגילה:**

1. **הלקוח יוצר תיקייה\קובץ חדש\ה .**
2. **המערכת יוצרת את אותה תיקייה\קובץ, אם מדובר בתיקיית סנכרון ראשית אזי גם ייווצר לו רישום ברשימה.**

**חריגים: אין.**

**אלטרנטיבות: אין.**

**USE Caseשינוי שם:**

**תיאור: המשתמש משנה שם של קובץ\תיקייה בתיקיית הסנכרון.**

**תנאי קדם: השרת כבר סנכרן אל המשתמש את תיקיות הסנכרון הראשיות.**

**תנאי סופי: השרת ימחוק בזיכרון שלו את אותו קובץ\תיקייה שמחק המשתמש.**

**זרימה רגילה:**

1. **הלקוח יוצר תיקייה\קובץ חדש\ה .**
2. **המערכת יוצרת את אותה תיקייה\קובץ בשרת, אם מדובר בתיקיית סנכרון ראשית אזי גם ישתנה לו הרישום ברשימה.**

**חריגים: אין.**

**אלטרנטיבות:אין.**

**USE Case עדכון:**

**תיאור: המשתמש משנה תוכן של קובץ בתיקיית הסנכרון.**

**תנאי קדם: השרת כבר סנכרן אל המשתמש את תיקיות הסנכרון הראשיות.**

**תנאי סופי: השרת יעדכן בזיכרון שלו את אותו קובץ שעדכן המשתמש.**

**זרימה רגילה:**

1. **הלקוח מעדכן קובץ כלשהו בתיקיית הסנכרון.**
2. **המערכת יוצרת את אותו קובץ בשרת.**

**חריגים: אין.**

**אלטרנטיבות: אין.**

3.2 דרישות של ממשקים חיצוניים

**3.2.1 ממשקי משתמש**

במערכת הלקוח קיים GUI

**3.2.2 ממשקי תקשורת**

קיימת תקשורת בין הלקוח לשרת

* 1. דרישות לא פונקציונליות

**בעת הסנכרון הקבצים יכווצו על מנת לאפשר תעבורת רשת יותר מהירה.**

3.4 דרישות בסיס נתונים

**אין בסיס נתונים לשרת. כל הנתונים נשמרים בקבצים הנמצאים בשרת.**