

1. מבוא

1.1 מטרה:

1 מטרת אפליקציית ShareIt היא ליצור פלטפורמה שיתופית בין סטודנטים באוניברסיטה למשלוחים פנימיים. האפליקציה מאפשרת לסטודנטים לבקש משלוחי מזון ומוצרים אחרים ולהציע שירותי משלוח, תוך שמירה על זמינות, פשטות ושיתוף פעולה קהילתי.

1.2 קהל יעד:

- **משתמשי קצה:**
 - סטודנטים המעוניינים לבצע או לקבל משלוחים בתוך האוניברסיטה.
 - חברי קהילות שמעוניינים בקבוצה שיתופית בין חברי הקהילה.
- **מפתחים:**
 - צוות הפיתוח של התוכנה, האחראי על המימוש והתחזוקה.
- **מנהלי פרויקטים:**
 - אחראים על פיתוח הפרויקט, לוחות זמנים ומטרות אסטרטגיות.
- **נותני חסות פוטנציאליים:**
 - משקיעים או ארגונים קהילתיים המעוניינים לתמוך בפלטפורמה.

1.3 שימוש מיועד:

תרחישים אופייניים כוללים:

- סטודנטים שמבקשים משלוחים יוכלו לפרסם בקשות ולבחור שליחים.
- סטודנטים שמבצעים משלוחים יוכלו לקבל בקשות ולבצע אותן לפי מיקומם הפנוי והעדפותיהם.
- האפליקציה תציע גם אפשרות לצבור קרדיטים על משלוחים.

1.4 תחום המוצר:

- **תכונות עיקריות:**
 - יצירת בקשות משלוח.
 - מעקב בזמן אמת אחר סטטוס המשלוח.
 - מערכת קרדיטים לתגמול שליחים
- **מגבלות:**
 - המערכת פועלת רק בתחום שטח מוגדר מראש
 - התממשקות עם שירותי תשלום חיצוניים תתאפשר רק בשלב עתידי
 - היענות מוגבלת מצד משתמשים

1.5 הגדרות ורשימת ראשי תיבות:

- **משלוח (Delivery):** העברת מזון או מוצר אחר בין משתמשים.
- **שליח (Courier):** משתמש שמבצע את המשלוח.
- **מערכת קרדיטים (Credit System):** מערכת לניהול תגמולים על ביצוע משלוחים.

2. תיאור כללי

2.1 צרכי משתמש:

- סטודנטים מבקשים שירות משלוחים פשוט ומהיר ללא תשלום בתוך הקמפוס.
- חברי קהילה שרוצים לתקשר עם חברי קהילתם ולבקש מהם משלוח מהיר ונוח.
- צורך בתקשורת נוחה בין משתמשים.
- אפשרות לקבל ולתת שירותים בצורה הוגנת ומאובטחת.

2.2 הנחות ותלויות:

- למשתמשים קיימת גישה למכשירים ניידים.
- חיבור אינטרנט זמין בכל רחבי הקמפוס.
- המערכת תשתמש ב-API למיקום.

3. תכונות ודרישות מערכת

3.1 דרישות פונקציונליות:

- מערכת רישום ואימות משתמשים.
- יצירת וניהול בקשות משלוח.
- מעקב בזמן אמת אחר מיקום השליח.
- ניהול קרדיטים ותשלומים.

3.2 דרישות ממשקים חיצוניים:

- **ממשק משתמש:**
 - אפליקציה תומכת באנדרואיד ו-iOS.
 - עיצוב נוח לשימוש עם גישה מהירה לפונקציות מרכזיות.
- **אינטגרציות צד שלישי:**
 - Google Firebase.
 - Firebase Cloud Messaging.
 - API לשירותי מיקום.
- **אחסון בענן:**
 - Firebase Realtime Database.

- אחסון נתוני משתמש מאובטח.
- סנכרון נתונים בזמן אמת.

3.3 תכונות מערכת:

- התאמה מבוססת מיקום.
- מנגנון דירוג למשתמשים.
- אפשרות לתקשורת ישירה בין מבקש המשלוח לשליח.

3.4 דרישות לא פונקציונליות:

• ביצועים:

- זמן תגובה קטן משתי שניות.
- תמיכה במספר משתמשים במקביל.
- עיבוד נתונים יעיל.

• יכולת גידול:

- יכולת להתרחב אופקית.
- תמיכה בבסיס משתמשים גדל.
- תשתית ענן גמישה.

• אבטחה:

- הצפנת נתונים מקצה לקצה.
- אימות משתמש מאובטח.
- מעקב אחר התנהגות משתמשים.
- אפשרות לפרסום אנונימי.
- יכולת לחסום משתמשים זדוניים.

• זמינות:

- 99.5% זמינות.
- דיווח שגיאות אוטומטי.
- טיפול שגיאות אלגנטי.

4. צרכים ספציפיים של משתמשי קצה

4.1 תיאור קבוצות משתמשים:

• משתמשים רגילים:

- סטודנטים שמבקשים או מבצעים משלוחים.
- חברי קהילה שמבקשים או מבצעים משלוחים.

• מנהלים:

- ניטור תוכן.
- ניהול דיווחי משתמשים.
- שמירה על תקינות הפלטפורמה.

4.2 דרישות ייחודיות למשתמשים:

- הגדרות התראות מותאמות אישית.
- אפשרויות פרסום אנונימיות.
- מערכת דירוג ואמינות משתמשים.
- תוכן מקומי והעדפות שפה.

5. ניהול שינויים

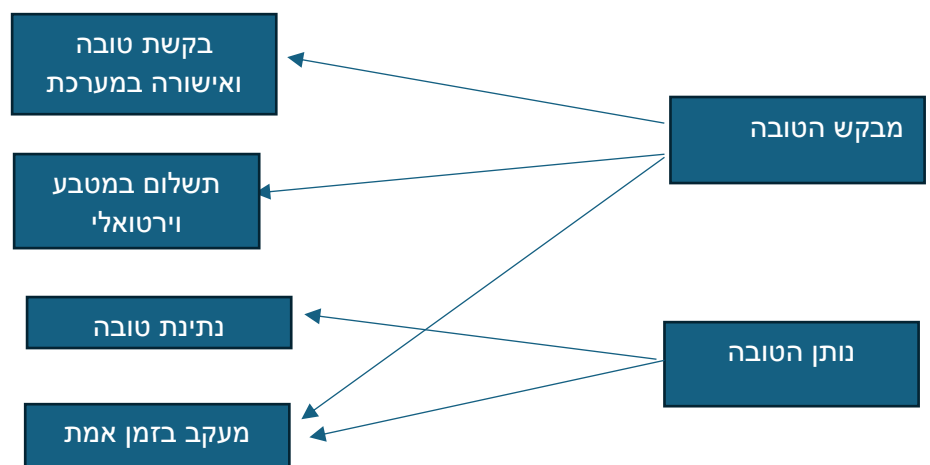
5.1 מעקב אחר שינויים:

- מסמך מבוקר גרסאות.
- תהליך סקירה משותף.
- נדרשת אישור מצד מנחי הפרויקט.
- יומן שינויים שקוף.

5.2 בקרת גרסאות:

- גרסה ראשונית: 1.0
- תאריך: [תאריך נוכחי]
- גורמים אחראיים: צוות פיתוח הפרויקט.
- מעקב אחר שינויים במסמך ייעודי להיסטוריית גרסאות.

דיאגרמת use-cases + הסברים:



Use Case 1: פרסום בקשת עזרה

מטרה: סטודנט רוצה לפרסם בקשת עזרה במערכת.

שחקנים מעורבים: סטודנט מבקש עזרה, מערכת.

תנאים מקדימים:

הסטודנט מחובר לחשבון שלו באפליקציה.

לסטודנט יש פרטי פרופיל מלאים.

שלבי התהליך:

הסטודנט לוחץ על כפתור "פרסם בקשה".

המערכת מבקשת ממנו למלא פרטים (סוג הבקשה, תיאור, מיקום, דחיפות, תמריץ מוצע).

הסטודנט ממלא את הפרטים ולוחץ על "אישור".

המערכת מאמת את הנתונים ומפרסמת את הבקשה במערכת.

סטודנטים מתאימים מקבלים התראה על הבקשה.

תוצאה מוצלחת: הבקשה מתפרסמת, וסטודנטים אחרים יכולים להיענות לה.

תוצאות חלופיות:

הסטודנט לא מילא את כל השדות הנדרשים – המערכת מציגה הודעת שגיאה.

הבקשה לא עומדת בקריטריונים (כמו כפילות או תוכן חסר) – המערכת מונעת פרסום.

Use Case 2: חיפוש והצטרפות לבקשה

מטרה: סטודנט רוצה לעזור לסטודנט אחר על ידי הצטרפות לבקשה קיימת.

שחקנים מעורבים: סטודנט נותן עזרה, סטודנט מבקש עזרה, מערכת.

תנאים מקדימים:

הסטודנט נותן העזרה מחובר לחשבון שלו באפליקציה.

שלבי התהליך:

הסטודנט מחפש בקשות עזרה על פי פרמטרים (כגון מיקום, סוג הבקשה, דחיפות).

המערכת מציגה רשימת בקשות עזרה רלוונטיות.

הסטודנט בוחר בקשה רלוונטית ולוחץ על "אני יכול לעזור".

המערכת מאמתת את הנתונים ומאשרת את המענה.

מבקש העזרה מקבל הודעה על סטודנט שנענה לבקשה.

תוצאה מוצלחת: סטודנט נותן העזרה מצטרף לבקשה, ושני הצדדים מקבלים עדכון.

תוצאות חלופיות:

אין בקשות זמינות התואמות לפרמטרים – המערכת מציגה הודעת "לא נמצאו בקשות עזרה."

מבקש העזרה ביטל את הבקשה – המערכת מעדכנת את נותן העזרה ומציעה חלופות.

Use Case 3: ביצוע תגמול ומעקב אחר התקדמות

מטרה: מבקש העזרה רוצה לתגמל את נותן העזרה ולעקוב אחרי התקדמות הבקשה.

שחקנים מעורבים: סטודנט מבקש עזרה, סטודנט נותן עזרה, מערכת.

תנאים מקדימים:

בקשת העזרה נענתה על ידי סטודנט נותן עזרה.

שלבי התהליך:

מבקש העזרה מקבל התראה לתגמל את נותן העזרה באמצעות מטבעות וירטואליים באפליקציה.

מבקש העזרה מבצע את התגמול בצורה מאובטחת.

נותן העזרה מקבל אישור על התגמול.

מבקש העזרה יכול לעקוב אחרי התקדמות בקשתו בזמן אמת (דרך מפה אינטראקטיבית או סטטוס משימה).

תוצאה מוצלחת: התגמול הושלם בהצלחה, ומבקש העזרה רואה את ההתקדמות בזמן אמת.

תוצאות חלופיות:

התגמול נכשל – המערכת שולחת הודעה עם בקשה לנסות שוב.

נותן העזרה לא ביצע את המשימה – המערכת מאפשרת לדווח על הבעיה או לשנות את הדירוג שלו.

מידע נוסף: מסמך ייזום פרויקט.