### מטלה 2 בהנדסת תוכנה: מסמך הדרישות

חברי הצוות:

1. ראש הצוות – שירה יוגב, מספר ת.ז. **325877108**
2. **שיראל כהן, מספר ת.ז. 324815083**
3. **דניאל בספלוב , מספר ת.ז.** 213446479
4. **מיכאל עידן, מספר ת.ז.** 315780130

### שוחחו עם לקוח פוטנציאלי

### במהלך הפגישה, הצוות ראיין את מיכאל עידן, חבר צוות הפיתוח של אפליקציית FoodShare ומשתמש פוטנציאלי. מיכאל הביע עניין רב באפליקציה וציין כי היא עונה על שני צרכים מרכזיים עבורו: האחד, האפשרות לתרום אוכל שהוא לא זקוק לו, והשני, הזדמנות לקבל אוכל בחינם, דבר שהוא רואה כיתרון משמעותי. מיכאל תיאר את עצמו כאדם שאוהב לחלוק מזון עם אחרים ובו זמנית נהנה למצוא פתרונות פשוטים ונוחים לשימוש באוכל שהוא זמין בקהילה. לדבריו, האפליקציה יכולה לשפר את היכולת לשתף מזון באופן יעיל ומאורגן.

### תמצית הדיון

### צרכים שזוהו:

### פלטפורמה שמאפשרת למשתמשים לפרסם ולמצוא פריטי מזון במהירות ובקלות.

### קטגוריזציה ברורה של סוגי מזון (למשל, מזון מבושל, מוצרי מזון יבשים, פירות וירקות).

### תפוגה אוטומטית של מודעות כדי לשמור על עדכניות ורלוונטיות בפיד.

### אתגרים קיימים:

### קושי למצוא פריטי מזון רלוונטיים במהירות בפלטפורמות אחרות כמו וואטסאפ או פייסבוק.

### חשש מבזבוז זמן בפנייה למשתמשים על מודעות שכבר אינן רלוונטיות.

### מאפיינים רצויים:

### ממשק פשוט ואינטואיטיבי לפרסום ולחיפוש מזון.

### אפשרויות לסינון לפי קטגוריות ומיקומים.

### התראות על פרסומים חדשים שמתאימים להעדפות המשתמש.

### כלים ניהוליים למנהלים כדי למנוע תוכן שאינו ראוי.

### מיכאל הדגיש את החשיבות של צמצום בזבוז מזון לצד יצירת תחושת קהילה שבה אנשים יכולים לשתף משאבים בקלות ובנוחות.

### 3) דיאגרמת Use case עם פירוט של ה- Use cases:

### בסוף, בחלק "הנספחים".

# Software Requirements Specification (SRS)

## Project Name: FoodShare - אפליקציה לשיתוף אוכל

# 1. Introduction

## 1.1 Purpose

### מטרת אפליקציית FoodShare היא ליצור פלטפורמה מסודרת ונוחה לשיתוף מזון בין חברי הקהילה. האפליקציה נועדה לצמצם בזבוז מזון, לספק דרך נוחה לפרסום ומציאת פריטי מזון, ולשפר את הנגישות למזון באמצעות תכונות יעילות וידידותיות למשתמש.

## 1.2 Intended Audience

### קהל היעד של האפליקציה כולל:

### משתמשי קצה: אנשים פרטיים המעוניינים למסור או לקבל מזון.

### עמותות או קבוצות קהילתיות המעוניינות לנהל שיתופי פעולה בתחום המזון.

### בתי עסק לממכר מזון (מאפיות וכד') שלא רוצים לזרוק את האוכל לפח האשפה.

## 1.3 Intended Use

### האפליקציה תשמש אנשים פרטיים לפרסום פריטי מזון זמינים, חיפוש מזון לפי קטגוריות, וניהול המודעות שהם פרסמו. מנהלים יפקחו על התוכן ויוודאו שהשימוש במערכת נעשה באופן הולם.

## 1.4 Product Scope

### האפליקציה מאפשרת למשתמשים:

### לפרסם מודעות על מזון זמין עם אפשרות לקבוע זמן תפוגה מותאם אישית.

### לחפש מזון באמצעות סינון לפי קטגוריות.

### לנהל מודעות אישיות (להעביר לארכיון, למחוק או לשחזר).

## 1.5 Definitions and Acronyms

### Firebase:

### מערכת שעוזרת לשמור נתונים, כמו מודעות של משתמשים, ולוודא שאנשים נכנסים עם פרטי התחברות נכונים.

### מנהל: משתמש בעל הרשאות מיוחדות לניהול תוכן ופיקוח על פעילות האפליקציה.

# 2. Overall Description

## 2.1 User Needs

### משתמשים זקוקים לפלטפורמה אמינה שתאפשר להם לשתף מזון ולמצוא מזון בקלות. האפליקציה נותנת מענה לצורך הזה על ידי מתן פתרון מסודר, יעיל ונוח לשימוש, שמאפשר לפרסם, לנהל ולחפש פריטי מזון בצורה פשוטה ומהירה.

## 2.2 Assumptions and Dependencies

### הנחות:

### למשתמשים יש גישה למכשירים ניידים עם חיבור לאינטרנט.

### מנהלי המערכת פועלים באופן פעיל לפיקוח על התוכן ולניהול תוכן שאינו ראוי.

### תלויות:

### השימוש ב- Firebase לניהול בסיס הנתונים ולאימות משתמשים.

### Android Studio משמש כסביבת הפיתוח של האפליקציה.

# 3. System Features and Requirements

## 3.1 Functional Requirements

### משתמשים יכולים ליצור, לערוך ולמחוק מודעות.

### המערכת תמחק באופן אוטומטי מודעות שפג תוקפן.

### מנהלים יכולים לחסום משתמשים ולהסיר תוכן שאינו מתאים.

## 3.2 External Interface Requirements

### ממשק משתמש:

### ממשק אינטואיטיבי ונוח לשימוש במכשירים ניידים.

### תמיכה בהתחברות באמצעות חשבונות Google ו.Facebook -

### שילובים חיצוניים:

### שימוש ב-Firebase לניהול פעולות ה- backend של האפליקציה.

### אחסון בענן:

### אחסון נתונים עבור מודעות וניהול משתמשים.

## 3.3 System Features

### מאפיינים מרכזיים כוללים:

### פרסום וניהול מודעות מזון.

### חיפוש מזון לפי קטגוריות.

### פאנל מנהל (מנהל אחד או קבוצת מנהלים) לפיקוח על התוכן, חסימת משתמשים המתנהגים בצורה שאינה הולמת וכו'.

## 3.4 Nonfunctional Requirements

### ביצועים:

### להבטיח שזמן התגובה לפעולות משתמשים לא יעלה על 2 שניות.

### סקלאביליות:

### אנחנו שואפים שהאפליקציה תוכל להתמודד עם עד 1,000 משתמשים פעילים בו-זמנית, מבלי לגרום להאטה בביצועים.

### אבטחה:

### אחסון נתונים ואימות משתמשים בצורה מוצפנת ומאובטחת. כדי להבטיח אבטחה גבוהה אנחנו מתכוונים לעשות את הדברים הבאים:

### *\* אחסון נתונים מוצפן:* כל הנתונים יישמרו בפורמט מוצפן, כך שגם אם גורם חיצוני מצליח לגשת לנתונים, הם אינם ניתנים לקריאה ללא מפתח פענוח.

### *\* אימות משתמשים:* אימות כניסה באמצעות Firebase , כולל אפשרות לכניסה דרך Google ו Facebook -, עם שכבת אבטחה נוספת (כמו אימות דו-שלבי בעתיד).

### *\* עדכוני אבטחה שוטפים:* בדיקות קבועות של פגיעויות ושדרוג המערכת בהתאם.

### *\* ניהול הרשאות:* הגבלת גישה לתכנים רגישים (סטטיסטיקות וכד') למנהלים בלבד.

### זמינות:

### הבטחת זמינות של המערכת ב-99% מהזמן.

### המערכת תהיה פעילה לאורך רוב הזמן, עם חלון זעיר לתקלות או תחזוקה (1% מהזמן, כלומר כ-7 שעות בחודש).

# 4. Specific Needs of End Users

## 4.1 Description of User Groups

### משתמשים רגילים: אנשים המעוניינים למסור מזון או לקבל פריטי מזון.

### מנהלים: משתמשים האחראים על ניהול התוכן באפליקציה, פיקוח על התנהלות שאינה הולמת וחסימת משתמשים שכותבים דברים לא הולמים. תהיה להם גישה לכל הסטטיסטיקות באפליקציה, ואת המסקנות מהן הם יוכלו להעביר לצוות הפיתוח.

## 4.2 Unique User Requirements

### משתמשים רגילים צריכים:

### ממשק פשוט ונוח לפרסום מודעות.

### אפשרויות חיפוש וסינון קלות לשימוש.

### מנהלים צריכים:

### גישה לכלים לניהול ופיקוח על התוכן.

### יכולת לחסום משתמשים ולהסיר תוכן שאינו מתאים.

### גישה לסטטיסטיקות באפליקציה.

# 5. Change Management

## 5.1 Change Tracking

### שינויים ינוהלו באמצעות GitHub שבו כל העדכונים יתועדו בהודעותcommit . כל חבר צוות יהיה אחראי לתאר בצורה ברורה את השינויים שביצע.

## 5.2 Version Control

### ניהול הגרסאות יתבצע באמצעות גיטהאב, תוך שימוש בתגיות (tags) וענפים (branches) לגרסאות מרכזיות, כנהוג בגיטהאב.

# 6. Appendices

## Use Case Diagrams

### כעת נציג את כל ה- Use Cases עם הדיאגרמות שלהם:

### Use Case 1: Posting Food

### מטרה: המשתמש מפרסם של מוצר מזון למסירה.

### שחקנים מעורבים: משתמש רגיל.

### תנאים מקדימים: המשתמש מחובר לחשבון שלו באפליקציה.

### שלבי התהליך:

### 1) המשתמש לוחץ על כפתור 'פרסום מודעה'.

### 2) המשתמש ממלא את פרטי המודעה (כותרת, תיאור, תמונות, קטגוריה).

### 3) המשתמש בוחר תאריך תפוגה למודעה.

### 4) המערכת שומרת את המודעה במסד הנתונים.

### תוצאה מוצלחת: המודעה מתפרסמת ומופיעה למשתמשים אחרים.

### תוצאות חלופיות: המשתמש לא מילא את כל השדות הנדרשים, והמערכת מציגה הודעת שגיאה.

### Use Case 2: Managing Content

**מטרה:**

המנהל מנהל את התוכן במערכת.

**שחקנים מעורבים:**

המנהל.

**תנאים מקדימים:**

המנהל מחובר למערכת עם הרשאות מנהל.

**שלבי התהליך:**

1) המנהל עובר לרשימת המודעות.

2) המנהל מזהה תוכן שאינו מתאים.

3) המנהל מסיר את המודעה או חוסם את המשתמש.

**תוצאה מוצלחת:**

התוכן הלא מתאים מוסר מהמערכת או שהמשתמש הסורר נחסם.

**תוצאות חלופיות:**

תמונה שמכילה טקסט, תרשים, צילום מסך, עיצוב

התיאור נוצר באופן אוטומטיהמנהל נתקל בשגיאה טכנית והמערכת מציגה הודעת שגיאה.

### Use Case 3: Searching for Food

**מטרה:**

המשתמש מחפש מזון לפי קטגוריות.

**שחקנים מעורבים:**

משתמש רגיל.

**תנאים מקדימים:**

המשתמש מחובר לחשבון שלו באפליקציה.

**שלבי התהליך:**

1) המשתמש פותח את מסך החיפוש.

2) המשתמש בוחר קטגוריות לחיפוש (למשל, מזון יבש, מזון מבושל).

3) המערכת מציגה את המודעות הרלוונטיות.

**תוצאה מוצלחת:**

המשתמש רואה רשימה של מודעות התואמות את החיפוש.

**תוצאות חלופיות:**

אין מודעות תואמות, והמערכת מציגה הודעה 'לא נמצאו תוצאות'.

תמונה שמכילה טקסט, תרשים, צילום מסך, קו

התיאור נוצר באופן אוטומטי