



המכללה האקדמית להנדסה ירושלים

פרויקט 3 – מפרט דרישות תוכנה (SRS)

מטרת המשימה

מטרת משימה זו היא להגדיר את מפרט דרישות התוכנה – Software Requirements Specification – של הפרויקט שלכם, מפרט שעליו יתבסס המשך הפיתוח.

הנחיות עבודה

כללי

ברכותינו! הצוות שלכם קיבל מימון לפרויקט שנבחר בשלב הקודם. הגורם המזמין היא חברת הון הסיכון הידועה סבב"ה (סיכויים בהנאה בבית הכרם) בע"מ. לחברה יש מנהלים (המתרגל והמרצה) ולקוח (שאתרתם בשלב האתחול). החברה והלקוחות ייפגשו אתכם מדי פעם לדון במוצר המבוקש. במשימה זו, הצוות שלכם יגדיר את הדרישות העיקריות למוצר התוכנה וכן יפרט דרישות סביבה של התוכנה עם סקיצות של חלק מממשק המשתמש.

דרישות חיצוניות

אמנם הלקוח התלהב מאד מהמאפיינים הכלליים של המוצר שהוצגו לו, ולכן גם החליט להשאיר בידיכם את השלב הבא להגדרה, אך עדיין יש לו כמה דרישות כלליות:

- המוצר צריך להיות קל לשימוש ככל האפשר, גם למי שאינו משתמש מחשב מנוסה (למעט כלים שנועדו עבור משתמשים כאלו).
- המוצר צריך להיות עמיד לשגיאות שאפשר לצפות שיקרו משימוש סביר, כמו קלט משתמש לא נכון, בעיות תקשורת וכדו'
- כמו מקודם, המוצר צריך להיות מבוסס לקוח-שרת, או לפחות בעל ארכיטקטורה של כמה רכיבים משמעותיים שלפחות אחד מהם מכיל לוגיקה שאינה טריוויאלית.
- חלק הלקוח צריך להיות נגיש מאד למשתמש. אם המוצר מבוסס רשת (SaaS), אז צריכה להיות גישה פתוחה אליו באינטרנט דרך URL באמצעות דפדפן סטנדרטי. במידה וצד הלקוח הוא אפליקציה עצמאית אז תצטרכו לתת דרך פשוטה ללקוח להתקין ולהריץ או לגשת גם לחלק השרת. אפשר להניח שהמשתמש כבר התקין כלים וספריות נחוצים (כמו JVM או .Net Framework).
- קוד המוצר יהיה פתוח (open source), למעט אם הוסכם עם הלקוחות שלא כך יהיה) ואחרים גם יוכלו להוריד את הקוד שלכם ולבנות את המוצר. בנוסף יהיה תיעוד, כך שאחרים יוכלו בקלות להבין ולהרחיב את המוצר שלכם. התיעוד יכלול מסמכי עיצוב, בדיקות ודוחות של תקלות.
- היקף הפרויקט יהיה בהתאמה טובה למשאבים שקיבלתם.

מעבר לדרישות אלו, אתם יכולים להמשיך בתוכנית הפיתוח ולהתחיל בבניית מפרט הדרישות. מסמך מפרט הדרישות שייכתב יהיה בעצם החוזה שלכם מול סבב"ה המגדיר את המוצר המפותח. כתוצאה מכך עליכם להיות בקשר אם הלקוח תוך כדי התכנון כדי לוודא שהמוצר עונה על דרישותיו וצרכיו.

מפרט דרישות – Software Requirements Specification (SRS)

מסמך SRS הוא כלי ללכידת (איסוף) והבנת (ניתוח) דרישות המוצר. בהמשך מסמך זה תמצאו תבנית מסמך SRS שהותאמה עבור משימה זו. המטרה היא לתרגל כתיבת תרחישי שימוש מפורטים לשם

ניתוח דרישות המוצר, אך גם לאסוף את כלל סיפורי המשתמשים לרשימה שעליה יתבסס המשך הפיתוח. אם אתם מעדיפים לעבוד בויקי – שימו לב ליצור עמוד המרכז את כלל הסעיפים הנדרשים במשימה זו.

השפעה על תהליך הפיתוח: מסמך זה מגדיר את מוצר התוכנה שייבנה, הוא נסמך על הצעת הפרויקט ומסמכים נוספים שהתקבלו מהלקוח. המסמך מגדיר אוסף דרישות שעל המוצר לממש ולכן עליו מתבססים השלבים הבאים, כגון תכנון, מימוש ובדיקות. לעיתים מסמך זה יכתב לאחר מסמך דרישות מערכת, המהווה קלט לכתיבת מסמך זה (שם נוסף למסמך User Requirements Document).

הנקודות העיקריות להתייחסות במסמך הן:

- תיאור כללי של המוצר והפעם בדגש מיוחד על ההיקף הצפוי שלו.
- אוסף use cases (תרחישי שימוש) הכולל לפחות דיאגרמת סיכום אחת ותיאור מלא של שניים מתרחישי השימושים העיקריים במערכת (אפילו במוצר בינוני בד"כ מוגדרים עשרות תרחישי שימוש).
- כלל הדרישות הידועות יכוסו ע"י רשימת סיפורי משתמש (בהמשך יזנו כ- github issues)
- גם דרישות כלליות מהמוצר, שהוזכרו לעיל, צריכות לבוא לידי ביטוי ברשימה זו וכן דרישות לא-פונקציונליות כגון סביבה (חומרה, תקנים וכדו')
- אב טיפוס של ממשק המשתמש. א. קישור לעמוד ההפצה הנוכחי. ב. הכניסו (או קשרו ל) לפחות שתי דיאגרמות המכילות סקיצה של ממשק המשתמש של המוצר וכיצד עוברים ביניהן.
- רשימה של דרישות לא ברורות או סותרות שנשארו לאחר ניתוח כל המקורות שלרשותכם.
- רשות: טבלת דרישות כללית. לכל דרישה יצוינו: מספר הדרישה, סוג הדרישה (פונקציונלית או לא), מספר תרחיש השימוש או סיפור המשתמש שהיא נובעת ממנו (אם רלוונטי) וכן דרגתה: הכרחית, כדאית או אופציונלית.

מסמך זה יהיה הבסיס להמשך הפיתוח. שימו לב שהכוונה שמסמך זה יהיה נושם, יתכן שתצטרכו בשלבים מאוחרים לעדכן אותו. אנחנו ממליצים שהמסמך יוכן כעמוד ויקי – אך אם תעדיפו תוכלו להשתמש בתבנית שבהמשך ההוראות (ואז כדאי כבר לחשוב על ניהול גרסאות למסמך). פעמים רבות מסמכים אלו הם באורך מאות ואלפי עמודים, אך בהתחשב במשאבים ואופי הפיתוח האיטרטיבי המתוכנן לפרויקט שלכם, הוא צריך להיות בסביבות 10 עמודים.

הנחיות הגשה וציונים

הגשה: העלאה לאתר הפרויקט עד הזמן הנקוב את התוצרים הדרושים והגשה

עד להרצאה הבאה:

- הכנה ופרסום של גרסה מתקדמת של מסמך ה-SRS (50%).
- תיאור קצר בויקי של מצב אב הטיפוס וניסיונות אחרים שהתחלתם לבצע להנמכת סיכון משמעותי שעלה בהצעת הפרויקט (4%).
- רישום לסקר: במסגרת ההרצאה או התרגיל עם סגל הקורס (1%).
- הצגה בסקר של דיאגרמת תרחישים, תרחיש ראשי, סיפורי משתמשים וסקיצת UI, במסגרת הסקר דווחו גם על הניסיון הטכנולוגי שלכם או אב-טיפוס.

יומיים אחרי ביצוע הסקר:

- הגשת גרסה מלאה של SRS (35%)
- הגשת סיכום סקר דרישות עם הלקוח (בדומה לתבנית שבנספח למסמך זה, ראו [ויקי](#))


ותיעוד של תקשורת נוספת עם הלקוח כרישום שיחות, שרשורי מייל, תרשימים על נייר (סרוק אם צריך) ובעיקר חוות דעתו של הלקוח על הדרישות כפי שהתגבשו בשלב זה (10%).


הנחיות לשימוש בתבנית

לאחר ההוראות מופיעה תבנית שנועדה לחסוך לכם את המצאת הגלגל וכוללת את המרכיבים העיקריים שמסמך כזה בד"כ מכיל. ניתן להוסיף או להוריד חלקים לפי הצורך.

גופנים:

טקסט כזה (אדום, נטוי עם קו תחתית), יש להחליף בטקסט משלכם.

טקסט המתחיל עם הציור  מהווה הסבר למקומו של התוכן בתהליך הפיתוח, יש למחוק לאחר מילוי המסמך.

טקסט המתחיל עם הציור  ולא **בגופן הזה** (סגול נטוי) מהווה הוראות למילוי, שניתן למחוק.

מקורות לתבנית

- תבניות הפרויקט ReadySET <http://readysset.tigris.org> (רשיון קוד פתוח BSD)
- תבניות מלוות לספר "Software Project Survival Guide", McConnell S., <http://www.construx.com/Page.aspx?nid=253>
- תבניות בשימוש התעשייה בארץ.

קישורים להרחבה

- Amber, Quick [UML Use Case Diagrams](#) Tutorial
- Mike Cohn, [User Stories](#) (presentation)
- Assaf Stove, [Why Should I INVEST in User Stories?](#), Blog 2012
- Wake, [INVEST in Good Stories, and SMART Tasks](#), 2003
- Steve Yegge, [Business Requirements are Bullshit](#)
- Spolsky, [Painless Functional Specifications - Part 1: Why Bother?](#)
- 830-1998 — IEEE Recommended Practice for Software Requirements Specifications. 1998. <http://dx.doi.org/10.1109%2FIEEESTD.1994.121431>

בהצלחה!

מפרט דרישות תוכנה - SRS

מידע כללי

שם הפרויקט:	<u>שם הפרויקט\המוצר</u>
צוות הפרויקט	<u>שמות (או כינויים)</u>
מסמכים מצורפים:	
מסמכים קשורים:	<u>הצעת פרויקט, תוכנית פיתוח, דרישות, מערכת, חוזה, קישורים נוספים</u>

לעשות: מלאו את המידע מעל ומתחת, הוסיפו ומחקו שורות בהתאם לצורך. עדכנו את תוכן העניינים.

תוכן העניינים

4.....	מידע כללי
4.....	הסטורית שינויים
5.....	1.הקדמה
5.....	1.1 מטרה
5.....	1.2 היקף
5.....	1.3 מילון מונחים
5.....	1.4סקירה
5.....	2.תרחישי שימוש – Use Cases
7.....	3.סיפורי משתמשים – UserStories
7.....	4.דרישות סביבה
8.....	4.1דרישות חומרה
8.....	4.2דרישות תוכנה
8.....	4.3דרישות נוספות
8.....	4.4ממשק משתמש – אב טיפוס

הסטורית שינויים

גרסה	תאריך	תיאור השינוי	מקור השינוי
1.0	1/10/2010	גרסה התחלתית	לא ישים

1. הקדמה

1.1 מטרה

זהו את מוצר התוכנה שמפותח. באיזו סביבה הוא יפעל. מהי הבעיה שבא לפתור. הסבירו מה הוא יעשה ומה הוא לא יעשה. הקשר שלו למוצרים אחרים, תארו את היתרונות היחסיים שלו, היעדים והמטרות. זהו את הלקוח והמשתמש המיועדים. הצעת הפרויקט ותוצרי האתחול מכילים פרטים אלו גם, אולם ההנחה היא שבינתיים המשכתם בגיבוש הרעיון מאחורי המוצר. זהו גם את התיעוד (בעדיפות למקוון) שיימסר עם מוצר התוכנה.

1.2 היקף

תארו את גבולות המערכת, כולל דברים שאינם כלולים במוצר. תארו בקצרה את החומרה והתוכנה שהמוצר תלוי בהם או יפעל בעזרתם. במידה ויש פרמטרים משמעותיים של ביצועים או אמינות, תארו אותם גם כן.

1.3 מילון מונחים

חשוב שמתהחלה תהיה הסכמה על מונחים שונים הקשורים למוצר ולפיתוח שלו

הכניסו כאן הגדרות וקיצורי דרך המשמשים בהמשך המסמך

1.4 סקירה

הכניסו כאן סקירה קצרה של תוכן המשך המסמך והתייחסות למסמכים קשורים

2. תרחישי שימוש – Use Cases

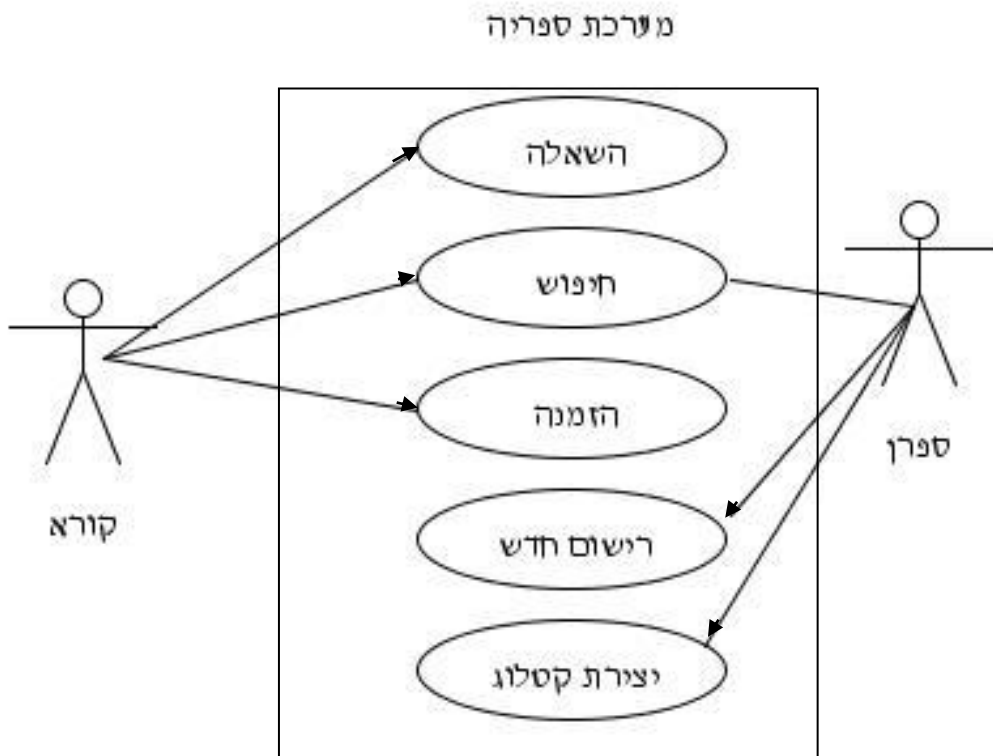
תרחישי השימוש מהווים כלי לניתוח הדרישות והבסיס לטבלת הדרישות שעל בסיסה ימשיך פיתוח המוצר.

ספקו לפחות:

1. טבלת שחקנים ובעלי עניין ומטרותיהם

שם שחקן	מטרות

2. תרשים סיכום UML-I של תרחישי השימוש במערכת
(אחד לפחות), הכולל את התרחישים העיקריים והשחקנים המשמעותיים במערכת



3. תרחישי שימוש פורמאליים עיקריים
עבור שני תרחישים שנראים לכם החשובים ביותר במערכת (התייעצו גם עם הלקוח, אחד מהתרחישים צריך לכלול את הסיכון העיקרי שזיהיתם ולאו החידוש העיקרי שבמוצר). הם צריכים להיות בסגנון של הדוגמאות שראינו (ראו גם במקורות ההרצאה, למשל Cockburn), במיוחד יש לכלול: שחקן ראשי, תנאי קדם, תנאי סיום מוצלח ונכשל, טריגר, רשימת צעדים של תרחיש ההצלחה, רשימה של הרחבות וטיפול בשגיאות בהתאם. ברור, שאי אפשר לחשוב מראש על כל התקלות האפשריות והפתרונות שלהם, ובכל זאת רשימת ההרחבות צריכה לשקף מחשבה מעמיקה על התרחיש. אם לתקלה מסוימת אין כרגע מענה, יש לציין זאת, ולהסביר מה בכוונתכם לעשות בהמשך. אם יש צורך אפשר לקשר את התרחיש לדיאגרמת פעילות או מסך משתמש.

להלן תבנית לתרחיש עם דוגמא:

שם התרחיש	UC1. הזמנת ספר
שחקן ראשי	קורא
מטרה	קורא מעוניין לשריין ספר מתוך הקטלוג המקוון
היקף ורמה	מערכת הספרייה, משתמש
בעלי עניין ואינטרסים	קורא – לשריין ספר בעל הספרייה – שרות מורחב לרווחת הלקוחות
תיאור	(אם יש צורך, שניים שלשה משפטים המתארים את התרחיש)
טריגר	הקורא נכנס למערכת
תנאי קדם	הקורא עבר את מסך ההזדהות (login)
תנאי סיום מוצלח	הספר שמור עבור הקורא
תנאי סיום כישלון	הספר אינו שמור

<p><u>1. המערכת מציגה מסך ראשי עם קטלוג ואפשרות חיפוש</u></p> <p><u>2. הקורא מזין את שם הספר</u></p> <p><u>3. המערכת מציגה התאמות עם מיקומם</u></p> <p><u>4. הקורא בוחר התאמה ובקשה לשמירה</u></p> <p><u>5. המערכת מאשר את ההזמנה ומציגה את הקטלוג בחזרה</u></p>	<p>תרחיש הצלחה עיקרי</p>
<p><u>3א המערכת אינה מוצאת את הספר</u></p> <p><u>3א.1. ...</u></p>	<p>הרחבות (שגיאות)</p>
<p><u>2. הקורא מזין מחבר או נושא</u></p>	<p>תרחישים חלופיים</p>

3. סיפורי משתמשים – UserStories

סיפורי המשתמשים מהווים רשימה ברמה כללית של כלל דרישות המשתמשים במערכת הידועות כרגע. מטרתם לתעד את כלל היעדים השונים של הלקוח (כולל דרישות לא-פונקציונליות), באמצעות דוגמאות ברורות כך שניתן יהיה על בסיסם לתכנן ולהעריך את המשך הפיתוח.

כסו את כלל הדרישות הידועות מהמוצר, כולל אלו שפורט בתרחישי השימוש לעיל. מומלץ בשלב זה לתעד את הסיפורים בוויקי במקום במסמך כדי לעודד שיתוף סביבם (במקרה כזה יש להפנות ממסמך זה לדפי הוויקי המתאימים). דרישות פשוטות וברורות מספיק לכסות ע"י שמן (למשל login), אך הציפייה היא גם לדרישות שאותן יש להבהיר באמצעות דוגמאות, כדלקמן.

דוגמא:

סיפור	תיאור קצר
US1	<p><u>רכישת דיסק</u></p> <p><u>בתור חובב מוסיקה</u></p> <p><u>אני מעוניין לראות את הכותרים האחרונים</u></p> <p><u>כך שאוכל להזמין ולהנות מהדיסק של...</u></p>
US2	<p><u>כניסה לאתר</u></p> <p><u>בתור סטודנט</u></p> <p><u>אני מעוניין להיכנס לאתר הספרייה</u></p> <p><u>כך שאוכל להזמין ספרים מהבית</u></p> <p><u>הערות: יש להפריד בין נושא ה-login לבין פונקציונליות הזמנת ספר</u></p> <p><u>קריטריון קבלה: בהינתן הכתובת של הספרייה ושנעשה login מופיע מסך השאלת ספרים</u></p>
US3	<p><u>אבטחה</u></p> <p><u>כל המידע הקשור לרכישה יאובטח עם SSL</u></p>

4. דרישות סביבה

תארו את דרישות הסביבה של המוצר. דרישות אלו מתארות את הסביבה הרחבה יותר של המוצר, חומרה, תוכנה, נתונים (בסיסי נתונים) וכדו'. להלן מספר פסקאות לדוגמה:

4.1 דרישות חומרה

4.2 דרישות תוכנה

מוצרי תוכנה, שרותים, חבילות וממשקים (Application Programming Interfaces) שבתכנון לשימוש.

4.3 דרישות נוספות

תקנים נדרשים, אבטחה, אמינות, נוחות, עמידות לתקלות דרישות לא-פונקציונליות נוספות

4.4 ממשיק משתמש – אב טיפוס



ספקו תרשימים של הרעיון הכללי של ממשק המשתמש. אלו תרשימים של ממשקים עיקריים שיאפשרו למשתמש לבצע את התרחישים העיקריים (ציינו באילו תרחישים מדובר). יש צורך בלפחות שני תרשימים והקשר ביניהם. אפשר לצייר אותם ביד ולסרוק, ע"י תוכנת מחשב (תוכנה גרפית, PowerPoint), או לכידת מסך של תוכנת אב-טיפוס שהכנתם. אם ממסך מסוים נפתחת תיבת שיחה או גלילה משמעותית, כדאי לצרף גם אותן. אין צורך בתוצר יפה, אלא משהו משמעותי שמדגים חשיבה על האפשרויות שצריך לקבל המשתמש והמעברים בין המסכים. אם אב הטיפוס כבר מספיק מפורט מספיק לצרף צילומי מסך עבורו ובכל מקרה יש לתאר מה מצבו של אב הטיפוס



ממשקי המשתמש משלימים את תרחישי השימוש מכיוון שהם אלו שבעזרתם יפעילו השחקנים את המערכת.

5. רשימת דרישות לבירור נוסף



תעדו כאן דרישות ידועות (בד"כ ע"פ מסמך הגדרת הפרויקט) הסותרות דרישות אחרות או דרישות שיש לברר את היקפן

6. רשות: טבלת דרישות – User Requirements Definition / Backlog



טבלת הדרישות מקשרת בין צרכי הלקוח לבין מוצר התוכנה המפותח, היא נגזרת מתרחישי השימוש ומקורות נוספים. בשלב זה סיפורי המשתמש יכולים בד"כ להיכנס לטבלה כמו שהם. רשימה זו כוללת בד"כ פירוט ממוספר של כלל הדרישות הפונקציונליות והלא-פונקציונליות של המערכת (לעיתים מפרטים לאחר הטבלה כל דרישה בפסקת טקסט נוספת). כאשר מפרטים תתי משימות, מסמך זה גם נקרא לעיתים Work Breakdown Structure המשמש הלאה להערכה ותכנון.

בפיתוח בשיטת Scrum רשימה זו היא בעצם ה-Backlog של הפרויקט שממנו יילקחו דרישות למימוש עבור כל ספרינט. מכיוון שאנחנו מתכננים לעבור לפיתוח בשיטות אג'יל, שבהן הניתוח המפורט נעשה תוך כדי המימוש, חלק זה הוא רשות ואם בחרתם לא לעשותו נסתמך על רשימת סיפורי המשתמשים בלבד.



מלאו טבלת דרישות כדלהלן. לכל דרישה יש לפרט את שמה, סוג הדרישה (פונקציונלית, אמינות, שמישות, אבטחה, ביצועים, מדרגיות, תחזוקתיות וכדו'), מקור הדרישה (מס' תרחיש, מס' סיפור משתמש או מקור אחר), דרגת הדרישה (1- הכרחית, 2- כדאית או 3- אופציונלית) והערכת המאמץ למימושה. השאיפה היא לפירוט עד לרמה שיהיה ניתן להעריך את המאמץ לכל דרישה במספר ימי עבודה מועטים שלכם, כך שאם יש דרישה רחבה מדי כדאי לפרוט אותה לתת-דרישות.

כדאי לקבץ את הדרישות לקבוצות של דרישות קשורות שצפוי שהעבודה עליהם תהיה כנראה באותו שלב.

הנה דוגמת התחלה של רשימה לפרויקט יומן מקוון, הרשימה שלכם צריכה לכלול כ-20 מאפיינים.

מס'	דרישה\מאפיין	סוג	מקור	דרגה	מאמץ
1	<u>צור\מחק יומן חדש</u>	<u>פונקצ'</u>	<u>UC1</u>	<u>1</u>	
2	<u>צור\מחק אירוע חדש</u>	<u>פונקצ'</u>	<u>UC2</u>	<u>1</u>	
3	<u>צפה ביומן של חבר</u>	<u>פונקצ'</u>	<u>UC2</u>	<u>2</u>	
4	<u>הוסף אירוע ליומן חבר</u>	<u>פונקצ'</u>	<u>US1</u>	<u>2</u>	
5	<u>יבוא של יומן אוטלוק</u>	<u>פונקצ'</u>	<u>UC8</u>	<u>3</u>	
6	<u>מניעת כניסה למשתמש שאינו חבר</u>	<u>אבטחה</u>	<u>SOW</u>	<u>1</u>	

תבנית סיכום סקר דרישות תוכנה - SRS

Review

תעדו את מהלך הסקר. את טבלת המשימות נזין בהמשך באתר ניהול הפרויקט כ-issues. 

תוכן

שם הפרויקט הנסקר	
נציגי הלקוח (הסוקרים)	
מועד ומקום הפגישה	
שמות משתתפים	

נקודות שעלו בסקר

1.	<u>חסר תרחיש חלופי בתרחיש UC2</u>
2.	

טבלת משימות הנובעות מהסקר (אפשר לחלופין למלא במערכת המשימות **github issues**)

נושא	פעולה נדרשת	באחריות
<u>תרחיש UC2</u>	<u>הוספת תרחיש חלופי במקרה של הזנת פרטים שגויים</u>	<u>אלן טיורינג</u>