

בדיקת דרישות

shauli kirshenzweig 313551905
elad goldin 322517186

סיפור 1

דרישה 1:

When the board gets a login request with login data containing a username it will find a corresponding user in a database and return it.

- **Ambiguous:** לא
- **Testable:** כן, תחילה ניצור משתמש לצורך הבדיקה ואז נקרא לפונק' ההתחברות עם פרטי המשתמש שיצרנו ונצפה שתחזור תשובה חיובית להתאמת פרטי המשתמש
- **Feasible:** כן, נממש מחלקה ייעודית login handler אשר יקבל את פרטי המשתמש מהמשתמש ויחפש פרטים זהים להם בבסיס הנתונים. מימוש דרישה זו אמור לקחת כשעה של עבודה.
- הערה: לא ה-board אמור לבצע את פעולת אימות המשתמש אלא מחלקה ייעודית לכך בתוך הפרויקט.

דרישה 2:

When the board gets a user from the login, it will try and authorize the user using a provided email and iris scan.

- **Ambiguous:** כן, מהמשפט משתמע כי הכוונה היא שלאחר שהמשתמש הכניס פרטים ב-Login בכדי להתחבר הוא יצטרך גם לספק מייל וסריקת רשתית, אך ניתן להבין כי ההתחברות תעשה רק ע"י מייל וסריקת רשתית ללא צורך במתן פרטים קודמים.
- **Testable:** כן, ניצור משתמש ונספק למערכת מייל וסריקת רשתית המתאימה לו, לאחר מכן ננסה להתחבר עם משתמש זה בעזרת סריקת הרשתית והמייל בכדי לבדוק כי המערכת יכולה לבדוק אם הסריקה שסיפקנו בהתחברות מתאימה לזו השמורה בבסיס הנתונים.
- **Feasible:** לא, לפרויקט הזה חסר המשאבים, הזמן עבודה והידע הדרוש על מנת להשתמש באימות משתמש באמצעות סריקת רשתית.

דרישה 3:

When a user is successfully authenticated he is logged into his account.

- **Ambiguous:** לא
- **Testable:** כן, ניצור משתמש ולאחר מכן נתחבר אליו ונצפה כי לאחר שההתחברות תצליח flag המייצג האם משתמש מחובר או לא יהיה true.
- **Feasible:** כן, ניישם flag עבור המשתמש שכאשר הוא יתחבר בהצלחה ה-login handler ישנה אותו ל-true. דקות בודדות של עבודה.

דרישה 4:

When a user is rejected by the authentication system he will be notified of the rejection , returned to login screen and not allowed to log in.

- **Ambiguous:** לא ברור אם ה-authentication system הוא זה שיידע את המשתמש על הrejection או אם רכיב אחר במערכת אמור לעשות זאת.
- **Testable:** כן, נכניס פרטים של משתמש שאינו שמור בבסיס הנתונים של המערכת ונבדוק כי אכן לא מתאפשרת התחברות.
- **Feasible:** כן, נממש רכיב ב interface layer שכאשר הוא מקבל תשובה שלילית לבדיקת authentication הוא יקפיץ הודעת שגיאה ויחזיר את המשתמש למסך ההתחברות. כשעה עבודה

דרישה 5:

When given a register request with a username, email and iris scan a new user entry will be created in the database with default account settings and a the given login details.

- **Ambiguous:** לא
- **Testable:** כן, נבנה מחלקה ב business layer שמקבלת את הפרטים הדרושים, יוצרת משתמש וקוראת למחלקה המתאימה ב-DAL אשר תיצור מופע של הנתונים ב-DB, ניתן למחלקה הראשונה input חוקי מתאים ונוודא כי בסיום הפעולה נוצר מופע חדש בבסיס הנתונים.
- **Feasible:** לא, לא ניתן לממש מבחינת משאבים, זמן וידע שימוש בסריקת רשתית, גם כשהיא ניתנת לנו ב request צריך לדעת להשתמש בספריות המתאימות כדי לשמור אותה ולעבוד איתה.

סיפור 2

דרישה 1:

When a user selects a 'create new task' button a new task will be created with its creator identifier being the user.

- **Ambiguous:** לא
- **Testable:** כן, נלחץ על הכפתור ולאחר הלחיצה נצפה לראות כי המשימה החדשה נשמרה לבסיס הנתונים.
- **Feasible:** כן, ניצור כפתור ומחלקות המתאימות לו המתקשרות אליו בשכבות: DAL, business, interface שכאשר נלחץ על הכפתור המחלקה ב-interface תודיע על הלחיצה ב-business, שם תתבצע הפעולה הלוגית של יצירת משימה, והיא תישמר ע"י המחלקה ב-DAL לבסיס הנתונים. בין יצירת המחלקות המרובות ועבודה עם ה-DB ועם ה-GUI הדבר אמור לקחת כ-5 שעות ואף כ-7 עבור debugging של קוד בסדר גודל זה (במידת הצורך).

דרישה 2:

When a new task is created the user will be prompted to give the task a title and a summary.

- **Ambiguous:** לא
- **Testable:** כן, נלחץ על הכפתור ונצפה לראות כי עולה prompt המבקש את הכנסת הפרטים הנ"ל.
- **Feasible:** כן, נקשר פונקציה לכפתור שכאשר הוא נילחץ יופיע prompt (פשוט או חלון GUI) אשר יאפשר למשתמש להכניס את הפרטים הדרושים. כחצי שעה עד שעה עבודה (תלוי מימוש, למשל בניית חלון GUI ייקח זמן רב יותר).

דרישה 3:

When a task is committed it will be put into the "To do" column it's creation date will be set to the current time and its title will be bolded.

- **Ambiguous:** כן
- **Testable:** כן, ניתן ללחוץ על הכפתור ולוודא כי המשימה נמצאת בטור ה-todo, כי התאריך תואם את תאריך היצירה וכי הכותרת מודגשת.
- **Feasible:** כן, כחלק מיצירת המשימה לאחר הלחיצה על הכפתור נגדיר את הערך הדיפולטי של העמודה בה היא נמצאת להיות: todo ונגדיר את הזמן הנוכחי כזמן יצירת המשימה.

הדרישה בחלקה אינה פונקציונלית כיוון שהיא מתארת עיצוב גרפי של המערכת: "the title will be bolded"

דרישה 4:

When a task is selected by the user, the tasks creator identifier is compared to the user and returned.

- **Ambiguous:** מאוד, מה זה tasks creator identifier? איך משווים אותו לאובייקט מסוג משתמש, מה מוחזר, ע"פ מה נעשתה ההשוואה? לאיזה משתמש מתבצעת ההשוואה?
- **Testable:** לא, לא ברור מה צריך להיבדק וכיצד.
- **Feasible:** לא, מאותה סיבה.

דרישה 5:

When a task is compatible with the user that selected it. The task details will be displayed to the user.

- **Ambiguous:** לא
- **Testable:** לא, לא ידוע מה התנאי compatible בכדי לבצע את ההשוואה.
- **Feasible:** כן, ניצור מחלקה ב business layer אשר תבצע את ההשוואה, אם ההשוואה תצליח היא תבקש ממחלקה ב-DAL לשלוף את המידע על task מבסיס הנתונים, מידע זה יוחזר למחלקה הראשונה אשר תשלח אותו אל מחלקה ב interface layer שם אשר תציג את המידע הנ"ל ב-GUI. כשעתיים עבודה.

סיפור 3

דרישה 1:

When the user presses a button, the task checks if it is in the "done" column and returns True/False statement.

- **Ambiguous:** כן, איזה כפתור יגרום לפעולה זו? המשימה היא מופע של מחלקה, איך היא אמורה לעשות את הבדיקה? על איזה task תתבצע הפעולה?
- **Testable:** לא, לא ניתן לדעת איזה כפתור מקושר לאיזה task על מנת שנוכל לדעת מי ה task עבורו נרצה לבדוק האם הוא ב done column.
- **Feasible:** לא, מאותה הסיבה.

דרישה 2:

When the task is allowed to move, the board compares the target tasks creation date to the tasks in the column it would be moved to and returns its designated position.

- **Ambiguous:** כן, ע"פ הנוסח אנו משווים בין משימות בטור לבין זמן יצירת משימה של המשימה אותה אנו מעוניינים להזיז. לא ברור מה המיקום שנקבע, איך או ביחס למה ואיפה.
- **Testable:** לא, לא ברור מה צריך לבדוק.
- **Feasible:** לא, מאותה סיבה.

דרישה 3:

When the designated position is determined the task is put into the next column in the designated position.

- **Ambiguous:** כן, ה-designated position של מי? מה קובע את ה-designated position, המשתמש או משהו במערכת? מה הכוונה שה-designated position היא determined?
- **Testable:** לא, לא ברור את מה או איך צריך לבדוק, מי הגורם המושפע וממה.
- **Feasible:** לא, מאותה סיבה.

דרישה 4:

When a task is advanced or edited it's "last modified" field will be updated.

- **Ambiguous:** כן, לא ברורה הכוונה ב-advanced.
- **Testable:** כן, ניצור task כלשהו, לאחר מכן נערוך אותו וניראה האם השתנה תאריך ה"עריכה אחרונה שלו". לא ניתן לבדוק את במקרה בו task is advanced כי לא ברור מה זה.
- **Feasible:** כן, ניצור מחלקה אשר מקבלת הודעה האם התנאי של edited or advanced מתקיים והיא תשלח הודעה למחלקה ב-DAL אשר תשנה בבסיס הנתונים את המידע הרלוונטי ב-task ותחזיר אותו למחלקה הראשונה אשר תחזיר אותו לinterface layer כדי שהמידע המועדכן יוצג ב-GUI. כשעתיים עבודה.

דרישה 5:

when the user is logged in all of the columns are displayed.

- **Ambiguous:** כן, לא ברורה הכוונה ב"כל העמודות", של מי? של כל המשתמשים?
- **Testable:** כן, ניצור משתמש, ננסה להתחבר איתו וניראה האם מוצגות כל העמודות שאנו מצפים לראות.
- **Feasible:** כן, ניצור דף ב-Interface layer בו עליו יוצג המידע על העמודות הרלוונטיות שיילקחו ע"י מספר מחלקות (בהתאם לשכבות) מבסיס הנתונים, לאחר קבלת אישור ממחלקת login handler כי המשתמש התחבר בהצלחה וכי יש להציג לו כעת את המידע הרלוונטי. כשעתיים עבודה.

סיפור 4

דרישה 1:

When the user wants to edit a task is prompted to enter new task details.

- **Ambiguous:** לא(אך יש שגיאת ניסוח-he is במקום is)
- **Testable:** כן. ננסה לשנות פרטי task שיצרנו ונצפה לראות חלון prompt שנפתח.
- **Feasible:** כן. כאשר נממש את האזור מתוכו מצופה לקרוא לפונקציית שינוי הפרטים, נוסיף לפני כן פקודה לפתיחת ה prompt ופעולה זאת תיקח (בהנחה ששאר החלקים שאינם קשורים לדרישה זו כבר מומשו) כמספר דקות בלבד. עם זאת, במידה ונצטרך לממש את כל החלקים הנדרשים(למשל חלון GUI) על מנת ליישם דרישה זו בפועל, המימוש ייקח בערך כ-5 שעות.

דרישה 2:

When a task receives new details the a new version of the task is saved into the database and displayed in the same place in the column

- **Ambiguous:** כן. לא ברור מתוך הדרישה מה קורה עם הדרישה המקורית (ללא השינוי). האם היא נמחקת? מוחלפת? או שבעת קבלת פרטים חדשים נוצר task חדש לחלוטין ולא משתנה כלום task המקורי. (כמו כן שגיאת ניסוח - the a)
- **Testable:** כן. ניצור task, נשנה את פרטיו, ונראה כי אכן נשמרים הפרטים המעודכנים בבסיס הנתונים. (בין אם task חדש ובין אם task המקורי, שפרטיו התעדכנו)
- **Feasible:** כן, בעת עדכון פרטי task נקרא גם ל data layer על מנת להכניס את פרטי task המעודכנים לבסיס הנתונים. בהנחה וכל בסיס הפרויקט כבר מומש, יישום דרישה זו ייקח כמה דקות. במידה ולא, נצטרך לבנות את המודלים השונים והאינטראקציות בהם, בפרט את כל המעברים משכבת הפרזנטציה עד לשכבת המידע, ודבר זה יכול לדרוש עבודה של מספר ימים (בערך 2-3 ימים).

(הדרישה למיקום task באותו המקום בעמודה אינה פונקציונלית).

דרישה 3:

When a request to view an older version is received, the requested task with requested version is returned from the database (if existent) and displayed.

- **Ambiguous:** כן (הניסוח בעייתי. what requested task?, An older version of what?)
 - **Testable:** ניצור משימה, ונעדכן אותה מספר פעמים, כדי לקבל מספר גרסאות. ניצור בקשה להציג את פרטי המשימה באחת הגרסאות הקודמות שלה, ונצפה להתאמה בין מה שיוצג לבין מה שהוכנס לפרטי המשימה באותה הגרסה.
 - **Feasible:** כן, נתחשב בזה בכך שנגדיר את המפתח של כל משימה להיות המזהה שלה + הגרסה שלה. כלומר כאשר נכניס משימה לבסיס הנתונים נשמור את המשימה ואת פרטיה, וכמו כן נשמור את הגרסה של המשימה. בקריאה לפונקציה נמצא בבסיס הנתונים את המשימה בעלת המזהה המבוקש, ואת מזהה הגרסה המתאים. בהנחה וכל שאר החלקים של הפרויקט כבר מומשו, יישום הדרישה תיקח בערך 3 שעות.
- הערה:** הדרישה מערפלת אפילו יותר את הדרישה הקודמת בה לא ברור מה יוצג בלוח עצמו. האם כל הגרסאות של כל המשימות יוצגו? רק העדכניות ביותר?

דרישה 4:

When a request to delete the task is received, the task is remove from the "to do" column

- **Ambiguous:** כן (ומה במקרה שה task אינו נמצא ב-to do?) (כמו כן גם כאן לא נאמר מה קורה לשאר הגרסאות של task)
- **Testable:** ניצור task ולפי דרישות הוא נוצר בעמודת to do. נמחק את task ונראה כי אכן נמחק מהעמודה הנ"ל. (בדיקה לא שלמה אך מכסה את הדרישה כולה)
- **Feasible:** כן. יישום הדרישה ייקח מספר דקות, נשלח בקשה שתגיע עד ל DAL, ותקרא לפונקציית מחיקת ההודעה.

הערה: בדרישה זו כמו גם בדרישות רבות בעבודה, נוסחה הדרישה בצורה " בהינתן X". ניסוח זה אמנם קביל, אך אינו מתאר בצורה הטובה ביותר את עבודת המערכת. ניסוח חלופי שיכול להיות יותר טוב הוא "כאשר ישות עושה X/מפעילה X/מבקשת X" וכו' ...

דרישה 5:

When a request to delete the task is received, check if the task in the "to do" column.

- **Ambiguous:** לא (שגיאת ניסוח - IS in the to do column)
 - **Testable:** לא. זאת מכיוון שלא נאמר בכלל מה נעשה פה. איך אפשר לבדוק משהו שלא מחזיר כל אינדיקציה לגבי פלט?
 - **Feasible:** כן, אך היישום של דרישה זו בפני עצמה מיותר ואף מגוחך (פונקציית void שרק בודקת אם העמודה בה המשימה נמצאת היא to do). מימוש ייקח מספר שניות.
- הדרישה לא פונקציונלית, ואפילו לא ממש דרישה. זהו רק חלק מתוך פונקציה שמתבקשת מתוך דרישות אחרות, אך הדרישה לא עומדת כדרישה בעצמה.

דרישות לא פונקציונליות

דרישה 1:

Task summary must be less than 200 words.

- **Ambiguous:** לא
- **Testable:** כן, ניצור task וננסה לכתוב לו summary בעל יותר מ-200 מילים בציפייה שהדבר לא יתאפשר.
- **Feasible:** כן, נוסיף תנאי if אשר יבדוק האם הקלט המיועד ל task summary מכיל מעל ל-200 מילים (199 רווחים או מקפים או טאבים). מספר דקות בודדות.

דרישה 2:

The database should support up to 1000 older versions of each task.

- **Ambiguous:** לא
- **Testable:** כן, ניצור task ונשנה אותה 1000 פעמים ונצפה ליראות כי כל שינוי נשמר, וכשנשנה אותה פעם נוספת שינוי זה לא ישמר (או שהשינוי הישן ביתר ימחק).
- **Feasible:** לא, אין למערכת את המשאבים בנחוצים לתמוך בפונקציונליות זו.