

## קורס יסודות הנדסת תוכנה תשפ"ה: הגדרת החלק המעשי בקורס

### 1 רקע

בקורס זה אנו נלמד פיתוח תוכנה בגישה הנדסית, בפרט נכיר וניישם מתודולוגיה עם דגש לפיתוח אג'ילי של מוצרים, ללא לקוח קיים. בפרויקט יבואו לידי ביטוי מושגים בסיסיים בתהליכי פיתוח תוכנה, הכרת שיטות ומתודולוגיות ליישום הידע, תוך מתן דגש על הנדסת דרישות – פרסונות, , תקשורת ועבודת צוות. נקודת המוצא רעיון שיתורגם לתהליכים עסקיים – סיפורי משתמש, דרישות (משימות) לפיתוח עד מימוש הרעיון למוצר מוגמר, העונה לדרישות שהוגדרו.

הפיתוח יעשה תוך שימוש ב-GIT תוכנה לניהול גרסאות, ותוכנת Jira לניהול שלבי הפרויקט כך שתוכל להתבצע עבודה יעילה כולל מעקב אחרי משימות והתקדמות.

בכל אחת מהמשימות שיתוארו בהמשך, לא מוגדרת דרישה לשפה \ סביבת תכנות ספציפית. עם זאת, שימו לב שבמהלך הפרויקט תדרשו לממש בדיקות יחידה כך שבבחירת הסביבה יש לקחת זאת בחשבון בעת בחירת הסביבה (לדוגמה C#). לכן מומלץ לבחור באחת מהשפות הבאות Python/Java script.

כל קבוצה תורכב מ-4 סטודנטים, שיבחרו את אחד מהפרויקטים המוצעים באמצעות רישום לקבוצות במודל. העבודה על הפרויקט תמשך לאורך הסמסטר החל בתהליך הנדסת הדרישות ולבסוף הפיתוח. המטרה היא עבודה לפי שלבי פיתוח מקובלים, מציאת הפתרונות הנכונים ביותר עבור כל בעיה, בין אם מדובר בשלב התכנון, עיצוב או מימוש. השנה הפרויקטים בקורס יתמקדו בנושאים הבאים:

#### 1. תכנון עירוני [בנוסף למי שיממש עם חיישנים עד 5 נק' להגשה הסופית]

ההתפתחויות הטכנולוגיות השפיעו בצורה משמעותית על האופן שבו ערים מתוכננות ומנוהלות. בתהליכי קבלת ההחלטות העירוניות מעורבים שחקנים רבים, כגון ראשי ערים, אזרחים, מתכננים ופקידי עירייה, כשכל אחד מהם מביא עמו רקע מקצועי וידע שונה על העיר. בפרויקט זה נעסוק בפיתוח כלים דיגיטליים שסייעו לשפר את תהליכי קבלת ההחלטות באמצעות הנגשה וחשיפה של תחומי ידע שונים בעיר. תהליכי הדיגיטציה מאפשרים הפצה ופתיחה של נתונים למקבלי ההחלטות בצורה רחבה ומהירה. כמה דוגמאות: משולחן חול מסורתי לשולחן אינטראקטיבי. במשך שנים רבות, תהליכי קבלת ההחלטות התקיימו סביב שולחן החול המסורתי, שאפשר לדעת היכן נמצאים הכוחות בשטח ולסמלך תרחישים שונים. בנושא זה ניתן ליישם בשני כיוונים. שילוב של חומרה עם תוכנה באמצעות חיישנים או איסוף ואנליטיקות של הנתונים.

#### a. מערכת לתכנון עירוני שולחן אינטראקטיבי

במסגרת הפרויקט תפותח תוכנה לשולחן אינטראקטיבי לתכנון עירוני, עם מודולים/רכיבים (כגון לגו) שניתן להזיזם, שיאפשרו לבחון כיצד הוספה של מבנים תשפיע על מערך הנתונים העירוני. האינטרנט של הדברים (IoT) מאפשר לעיר להיות "חכמה" באמצעות ניטור ומעקב אחר הנעשה ברחבי העיר, תוך שימוש בחיישנים. קבוצה זו תעסוק בפריסה של סנסורים ברחבי העיר לניטור

הפעילות במרחבים העירוניים השונים, ובכך תאפשר קבלת החלטות מבוססות נתונים עדכניים מהנעשה בעיר. לצורך הפרויקט המימוש יתמקד כמובן באיזור מוגבל קטן שיכול להציג התכונות.

#### **b. עיבוד ואיסוף נתונים עירוניים**

בפרויקט זה יאספו נתונים מגוונים אודות היבטים עירוניים שונים בעיר החכמה. המטרה לייצר תובנות שונות מנתונים אלה ולהציגם למקבלי ההחלטות. בפרויקט זה ניתן להשתמש במאגרים חכמים קיימים בערים חכמות כגון <https://data.gov.il/organization/beer-sheva>

[/https://gis.br7.org.il/apps/br7](https://gis.br7.org.il/apps/br7)

## **2. בנית קהילה מרחבית**

קהילה היא יחידת המרחב הבסיסית הקטנה ביותר של העיר. עם זאת מחקר רבים מתייחסים למחסור ביכולת בכימות תמונות המרחב של הקהילות בעיר. המאפיינים של קהילה עשויים להיות מגוונים ורבים ובאמצעות כלי תוכנה חכם אפשר לייצר את הסיווג של האוכלוסייה לקהילה. מאפיינים הכוונה גם לנתונים אובייקטיביים כגון מיקום ג"ג, מצב משפחתי, תעסוקה, חינוך, מרכז חברתי, דתי או גם תפיסות סובייקטיביות. על סמך נתונים שנאספים ומשקולות שונים ניתן לסווג את האוכלוסייה לקהילות. מטרת הפרויקט הוא פיתוח כלי שיבצע את התהליך כדי לתת תמונת מצב למקבלי החלטות אילו קהילות קימות בעיר או במרחב וכו'.

## **3. מערכת טפסים כגון: בקשת סטודנט וטפסים נוספים לחנייה**

המערכת תאפשר לאסוף נתונים ולהגיש בקשות, כולל אישורן, הפקת דו"חות ואנליטקות. הטפסים צריכים לשקף צרכים אמיתיים של הסטודנטים במכללה לצורך ייעול התהליכים הקיימים.

## **4. מערכת תמחור**

בעולם בו תרבות הצריכה מפותחת ואנשים מבצעים פרויקטים בתחומים שונים, יש צורך במערכת התומכת בקבלת ההחלטות ומקבצת את ההיבטים השונים לצורך התכנון והביצוע. לדוגמה רכישת מחשבים לארגון כולל ציוד היקפי, שרתים וכו' או פרויקט שיפוץ הכולל עבודות שונות (גבס, נגרות, חשמל וכו'). המטרה לאפשר להבין ולהגדיר את הצרכים ולתמחר בהתאם כולל חלופות.

## 2 ההגשות יתייחסו לשלבי הפיתוח הבאים<sup>1</sup>:

### 2.1 מסמך VISON ו Project Charter (הגשה 1)

מתייחס לייזום ואיתחול הפרויקט, על בסיס הגישה האג'ילית, יכלול את תיאור הצורך והתכונות, הסיכונים והסיכויים של המוצר אותו מפתחים. הנחיות במודל. את מסמך הייזום יש להגיש לפי המבנה של Project charter שהוצג בשיעור. שימו לב שעל מנת לאפיין את הבעיה בה מטפלים יש להיעזר בלפחות מאמר אקדמי אחד או מקור רשמי של נתונים (לדוגמה הלשכה המרכזית לסטטיסטיקה). לנוחותכם העלנו תבנית (ראה נספח 1 ייזום product charter בהמשך).

### 2.2 הנדסת דרישות (הגשה 2)

לאחר יצירת מסמך הייזום, נייצר מסמך דרישות שיכלול את דרישות הלקוח ויוגש ב JIRA . פיתוח הדרישות לפי העקרונות שילמדו בשיעור. יש לשתף בחשבון את צוות ההוראה (המיילים נמצאים באתר המודל, תיקיית "כללי על הקורס"). הדרישות יכללו תיאור התרחישים, וסיפורי המשתמש הרלוונטיים. התוצרים יוגשו לג'ירה ולמודל (וורד) בהתאם להנחיות שבנספח 4.2 הגדרת הגשת הדרישות.

### 2.3 תכן

על פי סביבת הפיתוח שתבחר הסטודנטים יפתחו בסביבת OO-בהתאם יעוצבו מסמכי מסכים המתארים ממשק המערכת וארכיטקטורה. אפשר להשתמש בכל כלי, מומלץ לנצל את הכלי FIGMA או Visual Paradigm - [הקישור](#)

### 2.4 מימוש

כתיבת הקוד עצמו.

### 2.5 בדיקות

מסמך המפרט את סוגי הבדיקות והיחידות הנבדקות בתוכנה. הנחיות הפורמט יינתנו בתרגול בהמשך. בנוסף, יינתן תרגיל בדיקות יחידה להגשה אישית

הגדרת הפרויקט כוללת את כל שלבי הפיתוח (בהדרגה לפי הסילבוס). התייחסות למשתמשים שונים, ממשק משתמש (אפשרי web).

====

ניתן לבחור בשפות הפיתוח הבאות: Python/C/C++ /html/android Java/Java script -- **הערה לא**

**ניתן להשתמש ב C#/PHP.**

יש לקחת בחשבון שמי שיבחר בשפה לא מוכרת יידרש ללמוד עצמאית את הנדרש לפיתוח כולל סביבת בדיקות יחידה. כמו כן מי שיבחר בסביבה לא ידידותית לבדיקות יחידה יאלץ לפתח בדיקות מותאמות.

---

שלבים 2.3-2.5 יפותחו לאורך הסמסטר אך יוגשו כחלק מההגשה הסופית.<sup>1</sup>

פיתוח אתר ולא מערכת לוקאלית מזכה בבנוס של עד 5 נק' לציון הסופי של הגשת הפרויקט. נושא הפרויקט יאושר על ידי המתרגלים ויפורסם במודל.

### 3 הגדרות כלליות לפיתוח הפרויקט

- לפחות 4 מאגרי מידע שיהוו את בסיס הנתונים שלכם. בסיס הנתונים יכול להיות קבצים, עבודה עם data base בענן יהיה בנוס (למשל fire base, sql server).
- לפחות 3 פרסונות-סוגי הרשאות (סוגי משתמשים במערכת). ראה סעיפים 3&4.
- לכל פרסונה יש לתאר תרחיש (scenario) אחד שיכלול פעולות מקצה לקצה. אפשר לפצל לשני תרחישים אם מורכב מידי. ראה סעיף 3.
- לכל פרסונה לפחות 2 פיצ'רים.
- יש להגדיר לפחות 10 סיפורי משתמש (תהליכים) לכל פרסונה-משתמש שמתוכם לפחות 3 שאילתות (דו"חות). פיצ'ר יכלול לפחות סיפור משתמש אחד.
- חובה לאפשר למשתמש לחזור למסכים הקודמים מכל מקום בתוכנית.
- סיפורי המשתמש כתובים על פי הפורמט שילמד בכיתה. לסיפורי המשתמש ייכתבו דרישות לפיתוח- מטלות בהתאם לנדרש.
- כיסוי מלא של כל סיפורי המשתמש והדרישות.
- הדרישות כתובות בצורה מלאה.
- פירוט כל פיצ'ר לדרישה עסקית (סיפור משתמש) לדרישות לפיתוח באופן היררכי.
- פיתוח web או מימוש מלא של חיישנים יזכה בבנוס.
- הבנוסים מתייחסים להגשה הסופית עד 5 נק'.

AGILE Project Charter Template

The name of the file should be: charter\_2023\_groupid#

**GENERAL PROJECT INFORMATION**

1. PROJECT NAME
2. PROJECT Members (team and course staff)
3. STAKEHOLDERS
4. EXPECTED START DATE
5. EXPECTED COMPLETION DATE
6. DATE for the last doc UPDATED

**PROJECT DETAILS**

7. Executive summary—product vision

Short paragraph that provides high-level explanation of the problem the project will address, as well as the project's purpose, goals, metrics of success, and expected deliverables. Use Moore's principles <For, Who, The, Unlike, Our Product> and mark them in the text.

8. Customers/potential persona/users- Who are the target customers and users-- personas?
9. Problem definition & Objectives (business and user)- verified by SMART. High level & specific
10. [Specific; Measurable; Agreed; Realistic; Time constrained.] Here you should include an academic reference to support the problem.
11. Existing alternatives, competitors
12. Expected Benefits-- Why should customers buy this product?

13. Scope

14. Milestones-road map

15. Development environment & tools- architecture diagram for final submission.

16. Success metrics (final submission)

17. Risks (specific to the project) **OPTIONAL**

מסמך זה פותח בהשראת מקורות מידע אקדמיים ומהרשת כולל product vision בספר של SOMMERVILLE, שיטת SIX SIGMA, נוהל מפת"ח, ותבניות באתר [smartsheets](https://smartsheets.com).

Hellas, M. (2015). Project charter, scope, and time management in the movie industry.

### **Tips to fill the document.**

- **Project Name:** Enter the project name.
- **Expected Start Date:** Note the project's expected start date.
- **Expected Completion Date:** Note the project's expected end date.
- **Project executive summary:** Provide high-level explanations of the problem the project will address, as well as the project's purpose, business case, goals, metrics of success, and expected deliverables. It may be easier to write this part at the end of this submission.
- **Project Scope:** Detail what is (and is not) included in the project scope, including interfaces.
- **Tentative Schedule-road map:** Provide start and end dates for the project's key milestones.
- **Benefits:** Specify the project's expected benefits and advantages compared to the current situation.
- **Risks, Constraints, and Assumptions (optional):** List any project risks, constraints, and assumptions so that project sponsors can see that you're adequately prepared. Risks should be specific to the project and contain severity and probability (0-1) and the calculated risk assessment.

חלק זה מהווה המשך לחלק הייזום – Project vision and chapter – ויש קשר הדוק בין המסמכים (ראה איור 1). הוא מכיל את החלקים הבאים (תרשים 1 מתאר את היחסים בין האלמנטים), שימו לב להיכן להגיש כל חלק. אם מצוין word הכוונה להוסיף למסמך הייזום וליצור מסמך אחד מרוכז להגשה זו. יש לשמור את הקובץ בפורמט הבא req\_2024\_groupid#

1. **Persona** [Word]- name and short description based on the initiation-project chart submission per persona. Notice that the name should be the same as in Jira.
2. **Scenarios** [Word] - per persona. Mark the actions described in the scenario.
3. **Features** [WORD+JIRA] - write in WORD the description of the feature including who are the personas and the corresponding user stories (ONLY the IDs) per each Feature (as appears in JIRA). Use tags of the features in the JIRA.
4. **User stories per each feature** [JIRA], including detailed design of the user story and tags for importance and features.
5. **Detailed requirements** [JIRA] will be numbered according to the user story ID and will cover frontend, backend requirements and unit testing.

An example for user story & detailed requirements:

User story#1 As a teacher I can add academic materials to each class that I teach so that students will be able to download the docs and practice after class.

1.1 In the class main page, a submission box will be located on the upper right corner only if the user is the teacher of the class.

1.2 only pdfs files can be uploaded.

1.3 the max space is 1GB.

1.4 max number of files is 100.

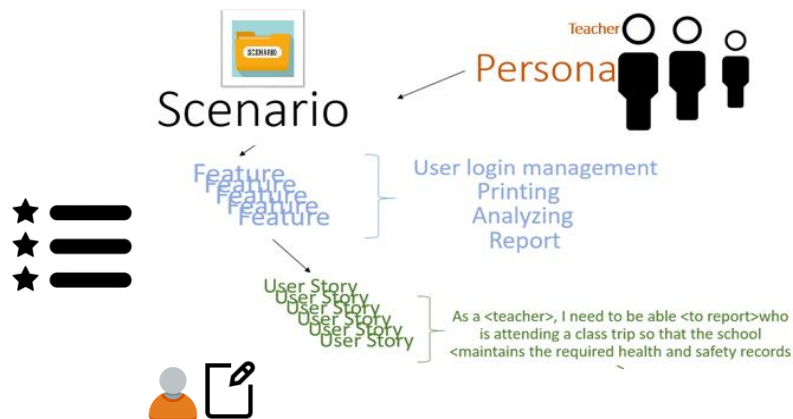
1.5 upload will be performed by clicking the button 'upload.'

1.6. a confirmation message will be displayed: "File <> was successfully uploaded.

1.7 Otherwise if the system encountered a problem a corresponding message will be : the upload of file <> failed due to the reason <>.

1.7 perform unit test to test the above detailed requirements.

- Use the tags to indicate the feature, persona and importance-priority of the user story.
- Verify that the user story is assigned to a specific team member.



איור 1 יחסים בין האלמנטים השונים בעיצוב בפרויקט אג'ילי

- לציין אילו פרסונות קיימות לדוגמה מורה, תלמיד, יועץ (הפירוט במסמך ייזום)
- תרחישים לפי פרסונה (בוורד) יש לתאר את סדר הפעולות ולהדגיש אותן. למשל:
- את הפיצ'רים יש לתאר בוורד בצורה של:

<action> the <results> <by|for|to><object>

למשל <users management>allows the users to access and logout the system --> אליו משוייכים סיפורי המשתמש #1,3,4. סיפורי המשתמש נמצאים בג'ירה.

- באיור מוצג סיפור משתמש עבור סטודנט, עם תגיות לחשיבות ולציון הפיצ'ר אליו שייך. כמו כן לשייך לחברי הצוות באופן מאוזן.

דוגמה למימוש סיפור משתמש. מצוין בכותרת המשתמש, סיפור המשתמש לפי התבנית, דרישות לפיתוח (detailed design) מפרטים איך סיפור המשתמש מעוצב. הערכת זמן לא ניתן להגדיר בשדות אלא רק בתיאור.

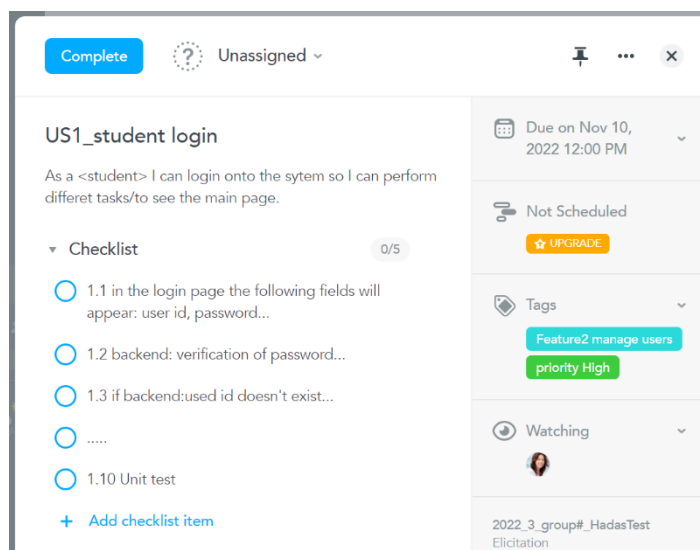


Figure 2 Defining elements in JIRA

**בהצלחה!**



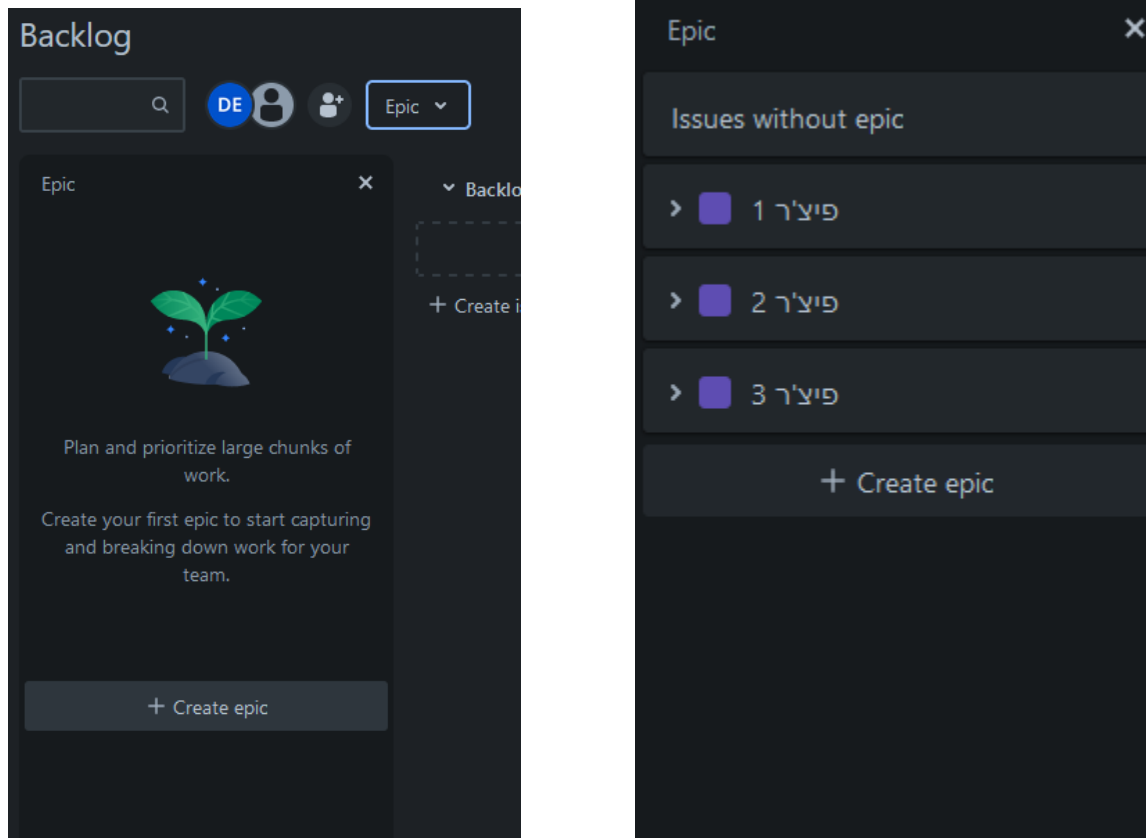
## 5 עבודה עם Jira

את הפיצ'רים, סיפורי המשתמש והדרישות לפיתוח אנו נגדיר בתוך עמוד Backlog.

### 5.1 הגדרת פיצ'רים

בJira, כל פיצ'ר נקרא Epic. על מנת להגדיר פיצ'ר חדש, יש לפתוח את תפריט Epic, ללחוץ על כפתור Create epic ולבחור בשם הפיצ'ר.

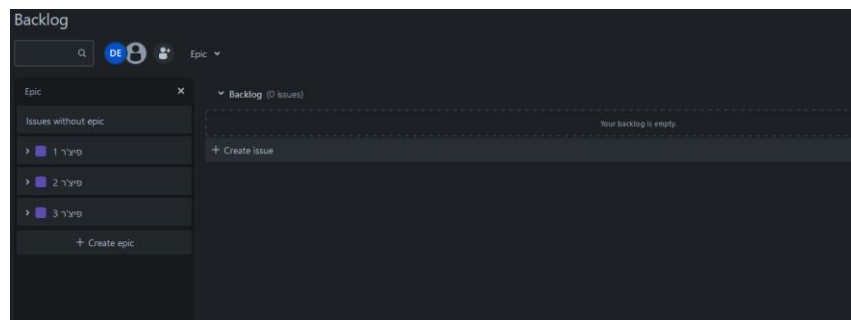
בדוגמא הבאה הגדרנו 3 פיצ'רים לפרויקט:



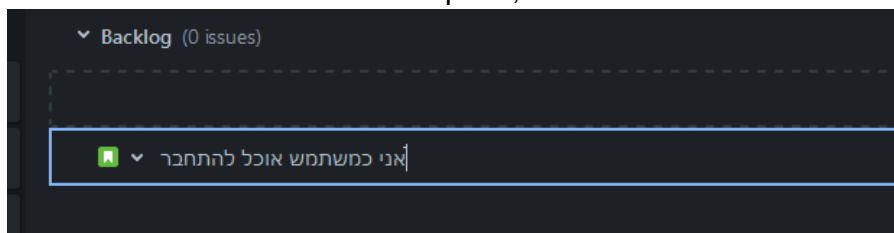
### 5.2 הגדרת סיפורי משתמש

בעמוד Backlog, מצד שמאל מופיעים הפיצ'רים שהגדרנו ומצד ימין מופיעים כל סיפורי המשתמש שנגדיר.

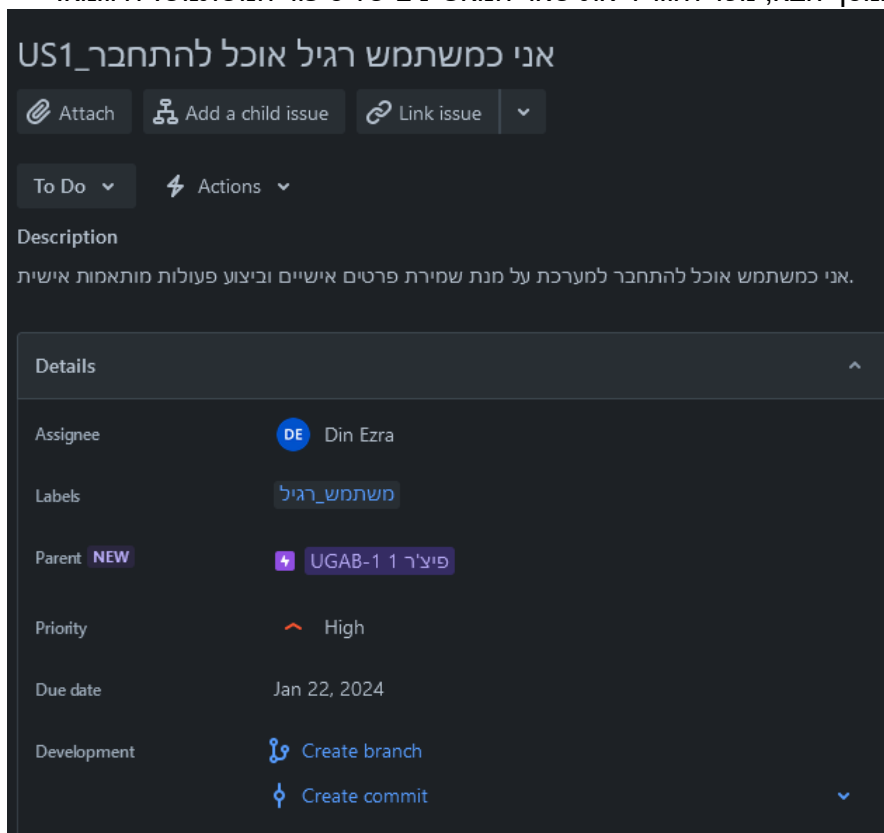
כדי להגדיר סיפור משתמש חדש, יש ללחוץ על Create Issue:



נרשום את כותרת סיפור המשתמש, ונלחץ Enter לשמירה:

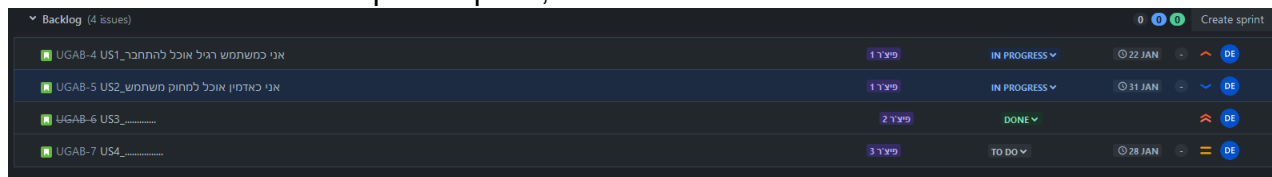


במסך הבא, נוכל להגדיר את שאר המאפיינים של סיפור המשתמש. לדוגמא:



בדוגמא הנ"ל, מתואר כותרת סיפור המשתמש, הסבר על הסיפור, מי מבצע, עדיפות, תאריך סיום מתוכנן ולאיזה פיצ'ר הסיפור שייך.

לאחר הגדרה של מספר סיפורי משתמש המשיכים לפיצ'רים שונים, המסך יראה כך:



### 5.3 הגדרת דרישות לפיתוח

כדי להגדיר דרישות לפיתוח, בתוך המסך של סיפור המשתמש הרלוונטי, יש ללחוץ על כפתור Add a child issue, ולתאר את הדרישות לפיתוח עבור אותו סיפור משתמש, לדוגמא:

אני כמשתמש רגיל אוכל להתחבר\_US1

Attach Add a child issue Link issue

In Progress Actions

Description

אני כמשתמש אוכל להתחבר למערכת על מנת שמירת פרטים אישיים וביצוע פעולות מותאמות אישית.

Child issues Order by 33% Done

UGAB-8	1.1 עיצוב עמוד ההתחברות	DE	DONE
UGAB-9	1.2 מתן אפשרות לקליטת שם משתמש וסיסמא	DE	DONE
UGAB-10	1.3 בדיקת הנתונים מול בסיס הנתונים	DE	IN PROGRESS
UGAB-11	1.4 הצגת הודעות שגיאה במקרה הצורך	DE	IN PROGRESS
UGAB-12	1.5 דרישות לפיתוח נוספת.....		TO DO
UGAB-13	1.6 דרישות לפיתוח נוספת.....		TO DO