

ניהול פרויקט תוכנה תשפ"א – הגשת הפרויקט

מספר קבוצה: 18

1. סיכום עיקרי הדברים:

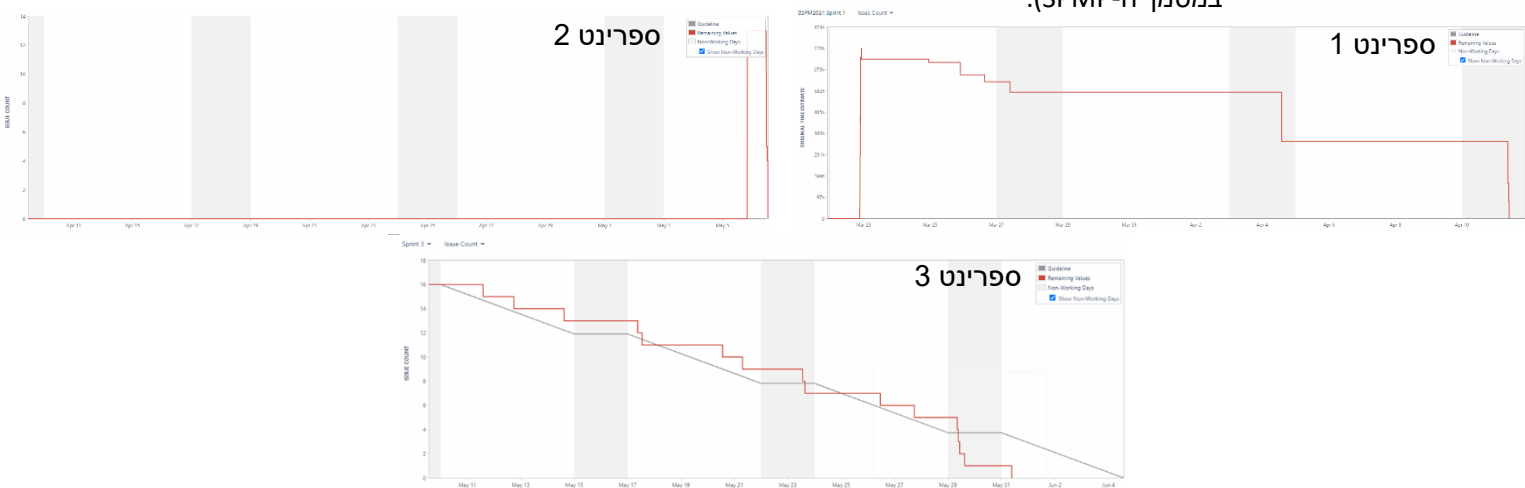
1.1. תקציר לפרויקט:

המערכת שלנו באה לתת פתרון לבעיה שנוצרה עקב סגירת מוסדות החינוך כתוצאה ממגפת הקורונה, כאשר אנשי החינוך ובניהם הגננות נאלצו לעבור ללמידה מרחוק. המטרה המרכזית של המערכת היא לאפשר לילדים בגילאי גן ללמוד מהבית באמצעות מערכת אינטראקטיבית ללמידה מרחוק ובאמצעותה לעודד למידה בקרב הילדים בעת שהותם בבית. המערכת שלנו כוללת מספר משחקים לימודיים המותאמים לילדי הגנים, הלימוד באמצעות משחקים אלו מקיף את נושאי הלימוד השונים ובכך מרחיב ומשפר את ידיעותיהם של הילדים בצורה ידידותית, וכמו כן, מאפשר לגננות לגשר על הפערים שנוצרו עקב הסגרים. במערכת שלנו, לגננת ניתנת האפשרות לערוך את תוכן המשחקים ובכך ליצור גיוון במשחקי המערכת. דרך המערכת ניתנת האפשרות להורי הילדים ליצור קשר עם המפקחת, ולדווח על בעיות במידה ויש. בנוסף לכך, במערכת שלנו ישנם דפי ניהול ופונקציונליות רחבה הן לגננת והן למפקחת אשר מאפשרים לנהל את הגנים בצורה הטובה והיעילה ביותר בתקופה זו, אם זה מבחינת מעקב אחרי התקדמות הילדים ואם זה מבחינת שיפור התנהלות הגנים באופן כללי, ובתקופת הקורונה בפרט.

1.2. בהתאם לג'ירה, סיכום העמידה בתוכנית העבודה (תכנון מול ביצוע) וכפי שהוצג ב-burndown chart:

בכל ספרינט הגדרנו WBS, ובו משימות ממוספרות באותיות אנגליות. כל משימה כללה סיפור משתמש אחד או יותר, וכל סיפור משתמש הוקצה עבור חבר צוות אחר. חלוקת העבודה הייתה שווה בין ארבעת חברי הצוות, דבר התרם לנו לעמוד בזמנים שהגדרנו לעצמנו בתרשימי GANTT ו-PERT עם תחילתו של כל ספרינט, וכמו כן, בזכות כך הצלחנו לממש את הדרישות על הצד הטוב ביותר. בחירת הדרישות העסקיות בכל ספרינט הייתה שקולה ומחושבת, בספרינט הראשון מימשנו את שלד המערכת, בהמשך, בספרינט השני מימשנו את לב המערכת ולבסוף, בספרינט השלישי והאחרון עיבנו את הפונקציונליות של כל אחד מהמשתמשים במערכת שלנו. הבחירה הקפדנית של הסיפורים בכל ספרינט, תרמה לנו לכבוש את היעדים שהצבנו לעצמנו ובזכות כך הגענו לתוצאות אליהם שאפנו. את העמידה בזמנים ניתן לראות בג'ירה, כאשר כל חברי הצוות הקפידו בכל תחילת עבודה על דרישה לפיתוח להעביר אותה ל-in progress, ועם סיום הדרישה וביצוע commit לגיטאהב, להעבירה ל-Done בג'ירה. בהתאם לכך, גרף ה-burndown chart בספרינט הראשון והשלישי הינו הדרגתי, מה שמעיד על עבודה שהתבצעה לאורך הספרינט ובאופן שוטף. יש לציין כי בספרינט השני, עקב בעיה כללית בשרת הג'ירה, ומחיקת הדרישות העסקיות ממנו, גרף ה-burndown שלנו איננו הדרגתי.

להלן burndown chart של שלושת הספרינטים שלנו (ניתן לראות את הגרפים בצורה ברורה יותר במסמך ה-SPMP):



1.3. מדדי עלות – הסבר + תוצאות ההרצה האחרונה

EVM (The Earned Value Management Metrics) הוא אוסף של מדדים אובייקטיביים ואמינים שניתן להשתמש בהם כדי לקבוע היקף, תקציב לאורך זמן והתקדמות לסיום. EVM מורכב מהערך המתוכנן (PV), הערך הנצבר (EV) והעלות בפועל (AC), ומאפשר להשוות במדויק ביצועים.

הערך הראשון של ערכי ה-EVM הוא ערך מתוכנן (PV). יש להשלים את המשימות שהקצנו בכל ספרינט המפורטות ב-WBS של כל ספרינט (ראה מסמך SPMP) בזמן הנתון. ניתן לקבוע PV על ידי ניתוח לוח הזמנים כפול האחוז המתוכנן של העבודה שהושלמה בתקציב הפרויקט.

העלות בפועל (AC) מחושבת לאחר תחילת העבודה. AC מוגדר כעלות הכוללת שהושגה בפועל בגין ביצוע עבודות עבור פעילות, AC הוא פשוט סכום הכסף שהוצאנו עד כה. לצורך כך צירפנו את העלויות הקבועות והמשתנות בפרויקט על-מנת להציג כיצד בוצעו חישובי העלות בפועל.

ערך נצבר (EV) מוגדר כערך העבודה שבוצעה, מבוסס במונחים של התקציב המאושר שהוקצה לאותה עבודה לפעילות. EV עוזר למצוא את ערך העבודה שהושלמה בפועל עד היום, ועוזר לענות על השאלות: האם אנחנו מתוזמנים? האם אנחנו במסגרת התקציב? כמו כן, אם ה-EV של הפרויקט נמוך מ-PV שלו, ניתן לומר כי אנחנו מפגרים בלוח הזמנים, אך אם ה-EV גדול מה-PV או שווה לו אנחנו עומדים בלוח הזמנים או מקדימים אותו. כמו כן, באופן דומה, ניתן להשוות את ה-EV ל-AC כדי לקבוע אם אנחנו מעל לתקציב הפרויקט או מתחת לו. ניתן לראות כיצד אלו באים לידי ביטוי בפרויקט בהמשך בטבלת החישוב ובגרף.

עלויות קבועות

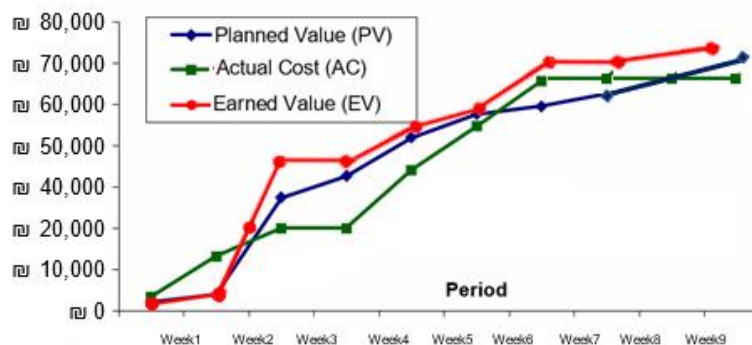
מה עולה	כמה עולה	סה"כ לספרינט 1	הסבר ספרינט 1	סה"כ לספרינט 2	הסבר ספרינט 2	סה"כ לספרינט 3	הסבר ספרינט 3	סה"כ לפרויקט
שעת עבודה למפתח לספרינט	100 ש	16,800 ש	4 מפתחים. הספרינט עמד על 3 שבועות (21 יום לספרינט). כל יום המפתחים השקיעו שעתיים כולל פגישות - $100*2*21*4$ – בהתאמה להיקף השעות בג'ירה.	20,000 ש	4 מפתחים. בשל הארכה לספרינט 2, נוספו 4 ימי עבודה לספרינט זה – סה"כ 25 ימי עבודה. כל יום המפתחים משקיעים שעתיים כולל פגישות - $100*2*25*4$ – (הערה: כרגע לא ניתן לבדוק את היקף השעות בג'ירה, בשל בעיית עדכון בשרת של הג'ירה)	21,600 ש	4 מפתחים. הספרינט עמד על 27 ימים. במשך 22 ימים מתוך הספרינט השקיעו בכל יום עבודה שעתיים כולל פגישות, בשאר הימים המפתחים הקדישו זמן לבדיקות יחידה ואינטגרציה, מילוי מסמכים	58,400 ש

	<p>– ופגישות 4*27*2*100 – ב-22 הימים הראשונים לספרינט, השעות באות בהתאמה להיקף השעות בג'ירה.</p>							
240 ₪	מחיר סטנדרטי לשירותי האינטרנט	80 ₪	מחיר סטנדרטי לשירותי האינטרנט	80 ₪	מחיר סטנדרטי לשירותי האינטרנט	80 ₪	80 ₪	שירותי אינטרנט (חודשי)
105 ₪	מחיר סטנדרטי לשירותי ענן	35 ₪	מחיר סטנדרטי לשירותי ענן	35 ₪	מחיר סטנדרטי לשירותי ענן	35 ₪	35 ₪	שירותי ענן (חודשי)
210 ₪	מחיר סטנדרטי לשירותי דומיין	70 ₪	מחיר סטנדרטי לשירותי דומיין	70 ₪	מחיר סטנדרטי לשירותי דומיין	70 ₪	70 ₪	דומיין (חודשי)
58,955 ₪								TOTAL

עלויות משתנות

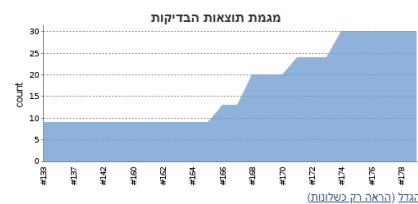
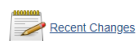
מה עולה	כמה עולה	סה"כ לספרינט 1	סה"כ לספרינט 2	סה"כ לספרינט 3	סה"כ לפרויקט
חשמל (חודשי)	300-400 ₪	400 ₪	400 ₪	400 ₪	1,200 ₪
קורסים אינטרנטיים בתשלום (תשלום חד פעמי)	50 ₪ לקורס	10 קורסים – 250 ₪	לא נלקחו קורסים בספרינט זה	לא נלקחו קורסים בספרינט זה	250 ₪
ציוד (חד פעמי)	2000 ₪ לציוד עבור מפתח 1	8000 ₪	לא התווסף ציוד נוסף מספרינט 1	לא התווסף ציוד נוסף מספרינט 1	8,000 ₪
גזברות (קפה, חד-פעמי, עוגיות)	100-300 ₪	150 ₪	150 ₪	150 ₪	450 ₪
ערב צוות	400 ₪ , 100 ₪ לחבר צוות 1.	לא התקיים ערב צוות בספרינט זה.	400 ₪	לא התקיים ערב צוות בספרינט זה.	400 ₪
TOTAL					10,300 ₪

כמה זמן לקח לבצע בפועל לעומת הזמן שתוכנן	69,255 ₪	AC (Actual Cost) = ACWP
(שערוך שלנו) המאמץ המתוכנן לכל יחידת עבודה לפי זמן	70,000 ₪	PV (Planned Value) = BCWS
עבור מקדמה - כמה משימות בוצעו ביחידת זמן (יותר או פחות ממה שתוכנן)	20,000 ₪	EV (Earned Value) = BCWP



תוצאות ההרצה האחרונה:

- מחק Pipeline
- הגדרה
- Move
- Full Stage View
- Job Config History
- Rename
- Pipeline Configuration History
- Pipeline Syntax
- GitHub Hook Log



המבדל (הראיה רק בשלמות)

Stage View

Build History	מצב
find	X
#180	11:21:03/06/2021
#179	21:46:02/06/2021
#178	21:45:02/06/2021
#177	21:15:02/06/2021
#176	21:15:02/06/2021

Average stage times:	
(Average full run time: ~1min)	
8s	
#180	No Changes
Jun 03	14:04

Declarative: Checkout SCM	Build	Metric1- Coverage	Metric2- PYLINT	Unit-Test
1s	23s	10s	23s	5s
992ms	28s	10s	23s	5s

תוצאות של המדד coverage המעריך את אחוז כיסוי הבדיקות במערכת שלנו על מספר שורות קוד בתוכנית:

+ coverage report --omit=/usr/*

Name	Stmts	Miss	Cover

atlas/__init__.py	0	0	100%
atlas/settings.py	32	0	100%
atlas/urls.py	5	0	100%
atlasapp/__init__.py	0	0	100%
atlasapp/admin.py	14	0	100%
atlasapp/apps.py	3	0	100%
atlasapp/choices.py	3	0	100%
atlasapp/forms.py	56	10	82%
atlasapp/migrations/0001_initial.py	8	0	100%
atlasapp/migrations/0002_auto_20210530_1815.py	7	0	100%
atlasapp/migrations/__init__.py	0	0	100%
atlasapp/models.py	76	4	95%
atlasapp/templatetags/SupervisorHome.py	13	9	31%
atlasapp/templatetags/__init__.py	0	0	100%
atlasapp/templatetags/bidudim.py	13	8	38%
atlasapp/tests.py	231	23	90%
atlasapp/urls.py	4	0	100%
atlasapp/views.py	357	273	24%
manage.py	12	2	83%

TOTAL	834	329	61%

תוצאות של המדד **pylint** המנתח את הקוד במטרה "לאכוף" סגנון קידוד של פייתון:

+ pylint --ignore=atlasapp/migrations --fail-under=-5 atlasapp

Your code has been rated at -0.77/10

1.4. תיאור כל הבדיקות שבוצעו בפרויקט – איכותית וכמותית תוך שימוש במדדים ועקרונות המופיעים ב-SPMP

לאורך הפרויקט ביצענו סדרת בדיקות, כאשר ביניהם: בדיקות יחידה, בדיקות פונקציונליות ולא פונקציונליות, בדיקות ETE, בדיקות ויזואליות ושמע, בדיקות ולידציה, בדיקות אינטגרציה וכמובן בדיקות מערכת.

כמו כן, כחלק מהבדיקות שביצענו השתמשנו במדדים כמותיים במטרה למדוד את איכות הקוד – המדדים שהשתמשנו בהם הם Coverage ו-pylint, כאשר Coverage הוא מדד המעריך את אחוז כיסוי הבדיקות במערכת שלנו על מספר שורות קוד בתוכנית, ו-pylint הוא מדד המנתח את הקוד במטרה "לאכוף" סגנון קידוד של שפת התכנות פייתון.

המטרה שלנו בשימוש במדדי בדיקות התוכנה הייתה לשפר את היעילות בתהליך בדיקת המערכת ולסייע לנו בקבלת החלטות טובות יותר להמשך תהליך הבדיקה על ידי מתן נתונים מהימנים.

בדיקות יחידה – מטרתם לבדוק כל חלק ולוודא שהחלק עובד באופן תקין. בפרויקט שלנו, בדיקות היחידה בוצעו בכל ספרינט עבור הפונקציות שבקובץ ה-views.py ועבור url-ים שבקובץ ה-urls.py. בעזרת בדיקות היחידה נוכל לוודא תקינותם של הפונקציות וה-urls. בסך הכל לאורך הפרויקט כ-80 בדיקות יחידה, כאשר בספרינט 2 מחצית מהם לא עברו, אך טופלו בהמשך, ואילו בספרינט 1 ו-3, 100% עברו. סך הכל בדיקות היחידה כיסו מחצית מהדרישות העסקיות בספרינט 1, כלל הדרישות העסקיות בספרינט 2, וכמו כן כלל הדרישות העסקיות בספרינט 3. בדיקות שלא עברו היו לנו לנוירה אדומה בנוגע לתקינות ה-Model ימים שלנו, הפונקציות וה-url-ים, וההשלכות הן ביצוע דיבוג לחלקי הקוד בהן הבדיקות לא עברו וטיפול בבאגים, מה שבאופן ישיר השפיע על כמות העבודה שיש לכל חבר צוות לבצע ובהתאם גם זמן. ניתן לומר שמדד ה-coverage שמימשנו בא לידי ביטוי בבדיקות אלו, שכן באמצעותו ניתן לדעת את אחוז כיסוי הבדיקות במערכת שלנו על מספר שורות קוד בתוכנית.

בדיקות מבוססות דרישות:

- בדיקות פונקציונליות - בדיקות המתקשרות ישירות לסיפורי המשתמש ולדרישות לפיתוח – בדיקות אלו התבצעו עם מימוש הדרישות לפיתוח, אין דרך להעריך זאת באופן כמותי כיוון שאלו בדיקות המתבצעות בעבודה השוטפת. בדיקות אלו כיסו את כלל הדרישות העסקיות בשלושת הספרינטים.
- בדיקות לא פונקציונליות - בדיקות שאין להן קשר ישירות לסיפורי המשתמש אלא עוסקות בביצועי התוכנית ובפרוטוקולים הנדרשים "מאחורי הקלעים" כמו ביצועים, אמינות ושמירות – בפרויקט שלנו, בדיקות אלו באו לידי ביטוי בבדיקת איכות הקוד, לבדיקות אלו לא היו זמנים מוגדרים מראש, אלא הם בוצעו לאורך כל ספרינט. ניתן לומר שמדד ה-pylint שמימשנו, מהווה מדד לבדיקות הלא פונקציונליות שלנו בפרויקט. כמו כן, בפרויקט שלנו באופן ספציפי, כישלון של בדיקות אלו לא מוביל להשלכות חמורות, כיוון שמדובר בסגנון קוד ונראות הקוד ופחות בביצועים ואמינות.
- בדיקות end to end - בדיקה המדמה מצב חי של המערכת במלואה – בוצע לאורך כל ספרינט. לא ניתן להעריך כמותית כמה מהן עברו או נכשלו, אך ניתן לומר שכאשר בדיקת E2E נכשלה, חבר הצוות שביצע את הבדיקה גם דאג לעלות על הבאג שגרם לבעיה וביצע בדיקה חוזרת עד שהבדיקה עברה בהצלחה. במידה ובדיקת E2E לא עברה, ההשלכות הן ביצוע דיבוג לחלקי הקוד בהן הבדיקה נפלה, ובהתאם טיפול בבאג, מה שבאופן ישיר השפיע על כמות העבודה שיש לכל חבר צוות לבצע ובהתאם גם זמן.
- בדיקות ויזואליות – בדיקות של האנימציות במשחקים – התבצעו לאחר מימוש המשחקים ומשיכת התמונות מבסיס הנתונים. לבדיקות אלו לא היו זמנים מוגדרים מראש, אלא הם

בוצעו לאורך כל ספרינט מכאן שאין דרך להעריך זאת באופן כמותי כיוון שאלו בדיקות המתבצעות כאשר נדרש בדרישה העסקית משיכת תמונות מבסיס הנתונים.

בדיקות שמע – בדיקות של קבצי השמע שהוספנו לאתר - התבצעו לאחר מימוש דרישות לפיתוח שדרשו שילוב של שמע ומשיכת קבצי השמע מבסיס הנתונים, גם בדיקות אלו מכאן לא ניתן להעריך באופן כמותי כיוון שאלו בדיקות המתבצעות כאשר נדרש בדרישה העסקית משיכת שמע מבסיס הנתונים.

בדיקות ולידציה – בדיקות שמטרתן לוודא שהמערכת עומדת בכל דרישות האיכות, היעילות והאבטחה שהוגדרו עבור המוצר הסופי. בפרויקט שלנו, בדיקות אלו באו לידי ביטוי בטופס ה-login, ובכללי בכל טופס (form) המחייב ולידציה, המפתח האחראי על הסיפור משתמש הכולל את הטופס דאג לביצוע ולידציות. גם בדיקות אלו מכאן לא ניתן להעריך באופן כמותי כיוון שאלו בדיקות המתבצעות כאשר נדרש בדרישה העסקית ליצור טופס. ההשלכות של בדיקת ולידציה שנופלת הן חמורות, שכן המערכת לא תעבוד כנדרש ובאופן חלק ללקוח במידה ובדיקת ולידציה אינה עוברת, לכן יש לטפל בבדיקה כזו שלא עוברת באופן מיידי.

בדיקות אינטגרציה – מטרתן לבדוק את ההשפעה שיש לחלק מוגדר במערכת עם חלקים אחרים במערכת ועם מערכות מקבילות ומשיקות. בדיקות אלו מבוצעות ע"י חיבור של כמה מודולים בודדים ומחברים אותם למערכת או לתת מערכת, או למערכת חיצונית. בדיקות האינטגרציה בפרויקט שלנו בוצעו בסוף כל ספרינט – בדיקות אינטגרציה ביצענו בספרינט 2 ו-3, כאשר בספרינט השני מימשנו 3 בדיקות, ובספרינט 3 מימשנו 5 בדיקות, כאשר 100% מהם עברו. בדיקות האינטגרציה חשובות על-מנת לדעת אם החיבור בין המרכיבים השונים של המערכת שלנו עובד באופן תקין.

בדיקות מערכת –בדיקת מערכת היא רמת בדיקה המאמתת את מוצר התוכנה השלם והמשולב במלואו. מטרת בדיקת המערכת היא להעריך את מפרט המערכת מקצה לקצה – בדיקת מערכת התבצעה בכל סוף ספרינט (סך הכל 3 בדיקות לאורך הפרויקט), לבדיקות אלו חשיבות רבה בזרימת המערכת וכמו כן טיפול בבאגים חדשים במידה וקיימים כאלו בעת בדיקת המערכת.

1.5. ניהול סיכונים עדכני לאחר ספרינט 3:

הסתברויות: סולם הנע בין 0 ל-1

- סבירות נמוכה - 0.1
- סבירות בינונית - 0.5
- סבירות גבוהה - 0.8

חומרה: סולם הנע בין 1 ל-8

- נזק קטן - 1
- נזק בינוני - 5
- נזק גבוה - 8

קריטריונים לסיווג רמת הסיכון: סולם הנע בין 0.1 ל-6.4
- מחושב על-פי המקרא למעלה (חומרה והסתברויות).

הערות:

- כל פגיעה כלכלית תהווה נזק גבוה, כולל ירידה במספר המשתמשים במערכת, הוצאות כלכליות החורגות מהתקציב ואי-עמידה בזמנים המהווה גם כן פגיעה כלכלית.
- כל פגיעה בחלק הניהולי של המערכת שלנו (ניהול הגן), יחשב לנזק בינוני
- כל פגיעה בחלק המעשיר והמלמד של המערכת שלנו (משחקים), יחשב לנזק בינוני
- פגיעה בשימוש במערכת כולה (הן ניהול הגן והן המשחקים), יהווה לנזק גבוה.
- כל פגיעה באבטחת מידע תהווה פגיעה חמורה באמון המשתמשים שלנו, ובכך להוביל לנזק גבוה

המאורע	סבירות	השפעה	רמת סיכון
מתחרים משיקים מערכת דומה הכוללת ניהול גן והעשרה ולימוד באופן מרחוק	סבירות גבוהה (0.8)	נזק גבוה (8)	<u>חישוב:</u> $8 * 0.8 = 6.4$
הגדרת יעדים לקוייה ומעקב לא אפקטיבי	סבירות נמוכה (0.1)	נזק גבוה (8)	<u>חישוב:</u> $8 * 0.1 = 0.8$
חוסר אינטגרציה בין צוות העבודה	סבירות נמוכה (0.1)	נזק גבוה (8)	<u>חישוב:</u> $8 * 0.1 = 0.8$
פיתוח של רכיבים לוקח יותר זמן מכפי שתוכנן (אי עמידה בזמנים שהוגדרו בתרשים GANTT)	סבירות גבוהה (0.8)	נזק בינוני (5)	<u>חישוב:</u> $5 * 0.8 = 4$
חזרה לשגרה בעקבות חיסונים, תגרום למערכת שלנו להיות לא רלוונטית	סבירות בינונית (0.5)	נזק גבוה (8)	<u>חישוב:</u> $8 * 0.5 = 4$

דלף של מידע על המשתמשים הרשומים באתר (כולל מפקחות, גננות ובמיוחד ילדים)	סבירות נמוכה (0.1)	נזק גבוה (8)	<u>חישוב</u> : $0.8 = 8 * 0.1$
פירצה למערכת של משרד הבריאות תוביל להשלכות על המערכת שלנו בעקבות שימוש ב-api שלהם	סבירות נמוכה (0.1)	נזק גבוה (8)	<u>חישוב</u> : $0.8 = 8 * 0.1$
עליה במספר הביקורות השליליות על המערכת מצד משתמשי המערכת	סבירות נמוכה (0.1)	נזק בינוני (5)	<u>חישוב</u> : $0.5 = 5 * 0.1$
עליה במספר הדיווחים על באגים במערכת מצד משתמשי המערכת	סבירות בינונית (0.5)	נזק בינוני (5)	<u>חישוב</u> : $2.5 = 5 * 0.5$
מחיקת DATA לא מגובה מבסיס הנתונים	סבירות נמוכה (0.1)	נזק גבוה (8)	<u>חישוב</u> : $0.8 = 8 * 0.1$
נפילת השרת עליו יושבת המערכת	סבירות בינונית (0.5)	נזק בינוני (5)	<u>חישוב</u> : $2.5 = 5 * 0.5$

1.6. דרישות עתידיות (שלא פותחו במסגרת הפרויקט):

- דיווח על באגים:
ממפקחת, אני רוצה לדווח על באגים במערכת שהורים / גננות דיווחו לי או שנתקלתי בעצמי, וזאת על-מנת ליעל את נושא הטיפול בתקלות וקיצור זמן הטיפול בתקלה.
דרישות לפיתוח:
 - Frontend - יצירת כפתור "פתיחת תקלה" ב-navbar בדף הבית של המפקחת
 - Frontend - יצירת דף "פתיחת תקלה":
 - כותרת h1 "דווח לנו על תקלה" / באג במערכת" בצבע אפור כהה
 - יצירת טופס בדף, ובו 2 input, כאשר אחד הוא נושא התקלה, והשני תוכן התקלה, ומתחת כפתור "דווח" בצבע אדום
 - יצירת חיבור ל-smtp של Gmail עם מטודות מובנות של Django. העברת המידע עם המטודות המובנות ושליחת מייל למייל המערכת.
- ביקורות משתמשים - כולל בתוכו 3 סיפורי משתמש:
 1. כגננת, אני רוצה לכתוב ביקורת למפקחת האזור על המערכת, וזאת על-מנת שאוכל לבטא את החוויה שלי מהמערכתדרישות לפיתוח:
 - יצירת Model Class חדש בשם "reviews" בקובץ models.py:
 - שדה 1 – header
 - שדה 2 – body
 - שדה 3 – rating
 - שדה 4 – מפתח זר, של User
 - שדה 5 – אובייקט DateTime
 - יצירת form חדש בקובץ forms.py התואם לשדות ב-Model "reviews".
 - Frontend – יצירת כפתור "כתוב לנו" ב-navbar בדף הבית של הגננת
 - Frontend – יצירת דף "כתוב לנו":
 - כותרת h1 "כתוב לנו את חוויתך מהמערכת שלנו" בצבע אפור כהה
 - יצירת טופס בדף, ובו 3 input, כאשר אחד הוא נושא, והשני הוא התוכן, והשלישי הוא דירוג (בין 1 ל-5). מתחת לאלו יהיה כפתור "שלח" בצבע כחול

- יצירת פונקציה בקובץ views.py השומרת את הנתונים שנשלחו בבסיס הנתונים.
- 2. כילד, אני רוצה שלהוריי תינתן האפשרות לכתוב ביקורת למפקחת האזור על המערכת, וזאת על-מנת שאוכל לבטא את החוויה שלי מהמערכת
דרישות לפיתוח:
 - Frontend – יצירת כפתור "כתוב לנו" ב-navbar בדף הבית של הילד
 - Frontend – יצירת דף "כתוב לנו":
 - כותרת h1 "כתוב לנו את חוויתך מהמערכת שלנו" בצבע אפור כהה
 - יצירת טופס בדף, ובו 3 input, כאשר אחד הוא נושא, והשני הוא התוכן, והשלישי הוא דירוג (בין 1 ל-5). מתחת לאלו יהיה כפתור "שלח" בצבע כחול
 - יצירת פונקציה בקובץ views.py השומרת את הנתונים שנשלחו בבסיס הנתונים.
- 3. כמפקחת, אני רוצה לקרוא ביקורות מהורים וגננות על המערכת, על-מנת שתהיה לי אינדיקציה על חוויות המשתמשים במערכת
דרישות לפיתוח:
 - Frontend – יצירת כפתור "צפה בביקורות משתמשים"
 - Frontend – יצירת דף "ביקורות משתמשים":
 - כותרת h1 "ביקורות משתמשים" בצבע אפור כהה
 - טבלה ובה הכותרות: "כותב הביקורת", "תאריך", "נושא", "תוכן", "דירוג כללי".
 - כתיבת פונקציה בקובץ views.py ובה משיכת הנתונים מטבלת "reviews".
- שימוש ב-api של משרד הבריאות להצגת נתונים:

כמפקחת, אני רוצה ש ב-dash-board שלי, יופיעו נתונים על מגפת הקורונה, וזאת על מנת להיות מעודכנת על מה שקורה בארץ בעניין נגיף הקורונה דרך המערכת, מבלי להיכנס לאתר נוסף.

דרישות לפיתוח:

 - Frontend - יצירת כותרת h1 מתחת לתוכן הנוכחי בדף הבית של המפקחת, "עדכונים שוטפים על נגיף הקורונה" בצבע אפור כהה.
 - Frontend - יצירת כותרת h5 מתחת לכותרת h1, "נתונים אלו נלקחו ממשרד הבריאות" בצבע אפור כהה.
 - Frontend – הצגת הנתונים מאתר משרד הבריאות בצורת גרפים וטבלאות.
 - Backend – שימוש ב-api של משרד הבריאות לצורך משיכת נתונים של משרד הבריאות בנושא נגיף הקורונה.

1.7. הפונקציונליות שלא מומשה אם קיימת, והסבר מדוע:

במסגרת הפרויקט עמדנו בכל הדרישות העסקיות שכתבנו עם תחילת הקורס. כלל סיפורי המשתמש, וכל הפונקציונליות שהתחייבנו אליה של שלושת המשתמשים במערכת שלנו, מומשו במלואם.

1.8. הנחיות להרצת המערכת, התקנות, אם צריך וקונפיגורציה:

1. `git clone https://github.com/snirben/Atlas.git`

2. `create virtual env (windows/mac/linux)`

3. `pip install -r requirements.txt`

4. Enter Folder atlas

5. `python manage.py makemigrations`

6. `python manage.py migrate`

7. `python manage.py runserver`

1.9. קישור לגיטאהב:

<https://github.com/snirben/Atlas.git>

2. סיכום ומסקנות – התרשמות כללית מתהליך הפיתוח, סיכום כמה דרישות פותחו, תובנות והפקת לקחים:

העבודה על הפרויקט הייתה מחד גיסא מאתגרת, ומאידך מלמדת ומעשירה. אחד האתגרים המרכזיים היה לדאוג שכלל העבודה על הפרויקט תהיה כהליך, ביצוע המשימות באופן הדרגתי וכמובן חלוקת עבודה שווה בין חברי הצוות. כמו כן, נדרשנו להקציב זמנים לכל משימה מראש ולהתחייב לעמידה בזמנים הללו שהקצבנו לעצמנו. למרות האתגרים, הפרויקט היה לנו לחוויה מלמדת ומעשירה. עבודת הצוות, שהיא חלק בלתי נפרד מעולם ההיי-טק, הייתה טובה, כל חבר צוות תרם מעצמו ונרתם למען הצלחת הפרויקט, במידה ואחד מחברי הצוות נתקל בקושי האחרים נרתמו לעזרתו, הקפדנו על פגישות צוותיות ובדיקת התקדמות של כל חבר צוות במשימות שהוקצו עבורו. כמו כן, דאגנו למתן code review, מה שהפך את הקוד לאיכותי יותר, וכמובן דבר זה היווה ביקורת בונה האחד לשני מה שעזר לכולנו ללמוד האחד מהשני, לצמוח ולהתפתח יחד.

בספרינט הראשון, החלטנו יחד עם המתרגלת שלנו מאי, לממש סיפורי משתמש המהווים את השלד של המערכת, וזאת על-מנת לא להעמיס על עצמנו בדריסת רגל ראשונה כצוות, והתנסות ראשונית עם המערכות החדשות כמו הג'ירה והג'נקינס. מבחינת בעיות לאורך הספרינט, לא עלו בעיות משמעותיות, פרט אולי להתקנת הסביבה שאצל חלק מחברי הצוות לקחה זמן בגלל בעיות רשת שנפתרו תוך זמן קצר. בדיעבד, אנחנו מבינים את החשיבות של בחירת הסיפורים בכל ספרינט, שכן הקפיצה למים העמוקים הייתה הדרגתית וכתוצאה מכך הצלחנו לעמוד ביעדים שהקצבנו לעצמנו בספרינט הראשון עם מינימום בעיות.

בספרינט השני לעומת זאת, דאגנו לממש דרישות עסקיות שהן high level, הן מבחינת הקושי במימוש הקוד, והן מבחינת חשיבות סיפורים אלו ללב המערכת שלנו. גם כאן, בחירת הסיפורים הייתה נבונה, שכן בסיום הספרינט השני המערכת שלנו קיבלה כמעט את צורתה הסופית, הפונקציונליות המרכזית בפרויקט מומשה, וזה הביא אותנו הרבה יותר מוכנים לקראת הספרינט האחרון.

בספרינט השני, פגשנו בעיות כאשר הבעיה המרכזית הייתה מחיקת הדרישות העסקיות מהג'ירה. כתוצאה מכך, נגרמה בעיה למי מאיתנו שנמחק סיפור המשתמש עליו עבד ולא יכל לבצע בשל כך Smart Commit. מכאן, שיש לחלקנו דרישות שאין סנכרון בין הג'ירה לגיטאהב.

כמו כן, בספרינט זה, עם קבלת המשוב להאקטון, בהמלצת המתרגלת שלנו החלטנו לעבור "לשיטה" אחרת מהשיטה שהתנהלנו בספרינט 1 בכל הנוגע לקומיטים ולעבודה עם הבראנצ'ים, מה שדרש מכלל חברי הצוות להתרגל באמצע הספרינט להתנהלות שונה מול הגיטאהב, ויצר התמודדות נוספת. לאחר שיחה מול המתרגלת החלטנו לחזור לשיטה הקודמת באישורה, וזאת מתוך ההבנה שהשיטה הקודמת בה עבדנו הייתה נוחה יותר עבורנו, ולראות עינינו גם יותר יעילה.

דבר נוסף, שהיווה עבורנו מכה קשה היה הפייפ ליין שלנו. לצערנו, לא הגדרנו בספרינט הראשון את הפייפ ליין להיות אוטומטי כאשר היינו בטוחים שכן, ובתום לב נוצר מצב שבספרינט השני, ברובו, לא התבצעה פעילות שוטפת בפייפ ליין מה שבא לידי ביטוי בהערכת הספרינט. בסופו של יום, ניתן לומר כי צלחנו את הספרינט השני, אם כי בקושי גדול יותר.

לספרינט השלישי, נותרו לנו לממש סיפורי משתמש שיהוו סגירת פינות של המערכת שלנו. סיפורים אלו היו בכללותם הרחבה למערכת הקיימת, ועיבוי הפונקציונליות של המשתמשים השונים במערכת. מבחינת בעיות שעלו בספרינט זה, הבעיה המרכזית הייתה האירועים הביטחוניים שהתרחשו, שכתוצאה מכך כלל חברי הצוות חוו טלטלה רגשית בגלל המצב מה שהקשה על כולם לקדם את הפרויקט, וכמובן הריצות אל החדרים המוגנים בכל אזהרה שקטעה לחלוטין את רצף העבודה כאשר מי מאיתנו עבד על הפרויקט באותו הרגע. עקב כך, היו לנו חששות שלא נצליח לעמוד בזמנים שהקצנו לכל סיפור משתמש בתרשימי GANTT ו- PERT שלנו, אך למרות הכל כלל חברי הצוות הפתיעו בנחישות ובהתמדה שלהם לבצע את המשימות שהוקצו עבורם על הצד הטוב ביותר, ולמרות המצב הנפשי והפיזי המטלטל, כולם עמדו בזמנים שהוקצו להם ושביעות הרצון שלנו מספרינט זה גבוהה.

תובנות שלנו, והפקת לקחים מהפרויקט -

עבודת הצוות היא דבר חשוב ומהותי בהצלחת הפרויקט. למרות שלפעמים עבודה בצוות יכולה להיות קושי, אפשר לצלוח את הקשיים אם ורק אם, יודעים להקשיב לאחר, לתת במה לרעיונות, ולהיות פתוחים לנסות הצעות של חברי צוות אחרים. אנחנו שמחים על שיתוף הפעולה הפורה בינינו ואנחנו בטוחים שזה היה גורם מרכזי בהצלחתנו בקורס.

על פי הניסיון שצברנו בעבודה על הפרויקט אנו ניקח איתנו הלאה את כתיבת הדרישות העסקיות טרם תחילת עבודה על פרויקט. דבר זה קריטי להצלחת הפרויקט, ומניף אותו להיות מדויק ובר ביצוע. כמו כן, גם חלוקת המשימות בין חברי הצוות, ועבודה לפי זמנים מוגדרים מראש לכל משימה, בעינינו זה דבר נפלא שעזר לנו להציג תוצר סופי על הצד הטוב ביותר בהתאם למגבלות הזמן והלחץ שהסמסטר מביא איתו. דבר נוסף שהקפדנו עליו, ולעניות דעתנו יש לשמר, זה עניין ה-code review שביצענו עבור כל commit של חבר צוות בגיטאהב. דבר זה מהווה ביקורת בונה חיובית, ובהכרח עזר לקוד שלנו להיות איכותי יותר ועם מינימום באגים. דבר חשוב לא פחות, שנשמח לקחת איתנו הלאה הוא חלוקת התפקידים לאורך הספרינט, Product Owner-I, DBA, Scrum master, שהייתה לנו לעוד מימד של התנסות ולמידה. כל אחד מאיתנו התנסה בתפקידים הללו בספרינטים השונים, ולדעתנו החלוקה הברורה הזו בכל ספרינט גם היוותה תרומה ענקית להצלחת הפרויקט. התפקידים הללו הם חלק חשוב ובלתי נפרד מפרויקט בממדים כאלו עם צוות.

ניתן לסכם ולומר, שלאורך הסמסטר, בכל הספרינטים, עמדנו במשימות שהקצבנו לעצמנו והצלחנו לממש את כל סיפורי המשתמש שהגדרנו בתחילת הסמסטר, 39 במספר. בסך הכל, אנחנו מרוצים מהתוצר הסופי שלנו, ואנחנו שמחים לקחת איתנו הלאה ארגז כלים חדש שיהווה לנו אבני דרך בכל פרויקט שניקח על עצמנו.