

הנחיות לביצוע פרויקט-ניהול פרויקטי תוכנה תשפ"א

במסגרת החלק היישומי של הקורס, יתבצע פרויקט פיתוח תוכנה בגישת OO, בשפת פייטון או java בלבד. סטודנטים שיחליטו למרות ההמלצה שלנו לעשות שימוש בשפות נוספות – יישאו באחריות בפיתוח Container/Docker מתאים לסביבה וכל הכרוך בכך. כל תהליך הפיתוח וניהול הפרויקט יתבצע באמצעות GitHub ו JIRA, תוך סנכרון ביניהם. המטרה לפתח את התוכנה בתהליך פיתוח אגילי בשיטת סקראם ובנוסף ליישם עקרונות CI/CD המדמה סביבת פיתוח בתעשייה, לאורך 3 ספרינטים על פי התאריכים המצויינים בסילבוס.

להלן דרישות כלליות לגבי הפרויקט. לכל דרישה יהיה ביטוי בשקלול הכולל של הפרויקט.

דרישות כלליות:

1. הגדרת כל הדרישות **כסיפורי משתמש** ו **Epics** ב **PBL** (Project BackLog) בתחילת הקורס- תכולה והיקף יאושרו על ידי צוות הקורס (שבוע 2). לכל ספרינט יוגדר **מראש** SPBL (Spring Back Log), כלומר תכולת הדרישות לאותו ספרינט (שבוע 2). יש להקפיד ולסיים את המתוכנן מבלי להעביר משימות לספרינט הבא. לכל סיפור משתמש יוגדרו משימות **tasks** ויושמו לאנשי הקבוצה על ידי הסקראם מאסטר. הסטאטוסים הקיימים Product Backlog, In progress, Design review, Done. לכל דרישה יש לוודא כיסוי של עיצוב ובדיקות ברמות השונות- ראה סעיף 6.
2. התיכנון של הפרויקט יתבצע באמצעות מסמך SPMP- שיעודכן על פי הצורך לאורך הסימסטר. יוגש בכל ספרינט.
3. התיכנון והביצוע בכל ספרינט ישתקף באמצעות **Burn down chart** כך שלאורך ההתקדמות יוספו הספרינטים לאותו חשבון. שימו לב יש לתחזק את התיכנון והעבודה באופן רציף לאורך הספרינטים ולא רק בסוף לצורך ההגשה.
4. הגדרת **בעלי תפקידים** – Jira. בכל ספרינט יתעדכן.
5. לכל ספרינט יוגדרו התפקידים כולל scrum master, על פי החלטת הקבוצה. כל אחד מחברי הקבוצה יאייש תפקיד נדרש כגון DBA, מפתח, עיצוב, לקוח בכל ספרינט. בכל ספרינט משחק התפקידים ישתנה כך שכל חבר בקבוצה יתנסה בתפקידים השונים.
6. כל צוות מחויב **להיפגש לפחות פעמיים** בשבוע ולדון על ההתקדמות, הבעיות, הפתרונות, מה עושים כעת ומה מתוכנן להמשך (טווח קצר). סיכומי פגישה יוגשו בפגישה עם המתרגל. בהתאם לעדכן את התכנון מול הביצוע. תיעוד המפגשים יוגש בתום הספרינט.
7. בכל ספרינט יתבצע **תכנון, review, ניהול סיכונים, עיצוב, פיתוח ובדיקות יחידה ואינטגרציה**- יתועד ב Jira באמצעות הגדרת משימות מתאימות.
8. מחזור חיי הפרויקט יכלול הגדרת מטלות **tasks** לכל סיפור משתמש (לפחות אחת). לכל מטלה יהיה אחראי שיוגדר על ידי ה scrum master. שלבי התקדמות של מטלות. הסטאטוסים הקיימים:

To do, In progress, Done

9. לכל מטלה הצוות אחראי להגדיר ב Github בדיקות מסודרות ברמת יחידה ותת-מערכת. הבדיקות יכללו בדיקות מערכת ואינטגרציה. השבוע האחרון יוקדש למקצה שיפורים ובדיקות סופיות.
10. יש לקשר את הקוד המתאים לכל סיפור משתמש בגיירה.
11. בסוף כל ספרינט יש להגיש עם מסמך ה SPMP, סיכום (ייצוא מ Jira) של Burn down chart, מסמך בדיקות ברמת יחידה ותת מערכת המתאר את הבדיקות הנדרשות לפונקציות השונות, ואת תוצאות הבדיקה כפי שהתקבלו בהרצה. סטאטוס ההרצה יופיע במטלה ב GIT. פורמט הבדיקות יכול לכלול את החלקים if, when, then.
12. תוכנית עבודה תוצג ב Jira Roadmap שב Jira.
13. בפגישת התכנון (בתחילת כל ספרינט) עם המתרגל, ה scrum master אחראי להגדרות מסודרות של כל המטלות בספרינט כולל השמה לסטודנטים והערכת זמנים. בפגישה יעברו על התכנון והמתרגל יאשר או יעדכן במידת הצורך. יש להקפיד להגיע מוכנים לפגישה.
14. בפגישת הסגירה (בתום כל ספרינט) עם המתרגל (retrospective) **sprint summary meeting**, ה scrum master אחראי לסגירה מסודרת של כל המטלות בספרינט כולל עדכון של תוצאות הבדיקות-מראש. סיכום בעיות, מצב נוכחי ותוכנית להמשך. יש להגיש סיכום לכל ספרינט (2-3 עמודים) לתיבת ההגשה. הסיכום יכלול 3 תתי סיכומים שהוצגו בפגישות השבועיות במהלך הספרינט (ראה סעיף 4).
15. Check in של הקוד **בסביבת הפיתוח יקושר לסיפור משתמש\issues**, כך שלכל דרישה יהיה כיסוי של קוד. כל חברי הצוות ייקחו חלק שווה בפיתוח.
16. בכל ספרינט יתבצע עיצוב של תרשימי UML, על פי ההנחיות שיתפרסו בהמשך ובתרגול (יתבטא בקישור ל Github). יוגש כחלק מהמסמכים בסוף הספרינט.
17. פיתוח אפליקציית מובייל תזכה בבונוס עד 5 נק' לציון הסופי.

תיאור הפרויקטים לפי הנושאים הבאים. שימו לב כי אנו מצפים לחשיבה יצירתית על מנת לפתח את המערכות על בסיס ההגדרות הכלליות המופיעות בהמשך.

לכל פרויקט יש ליישם:

1. לפחות 3 משתמשים\הרשאות.
2. בסיס נתונים : כגון : SQL Server , mongoDB.
3. לפחות 36 סיפורי משתמש (לפחות 12 לכל משתמש).
4. לכל משתמש לפחות 6 שאילתות שהוא יכול לבצע.
5. לפחות 2 הורשות.
6. על המערכת להיות יעילה ככל האפשר, כאשר המודולים הנבחרים מסונכרנים מבחינת המידע במערכת (לדוגמא: בנושא 1 יש לתכנן את שיבוץ הדרישות בתוכנית עבודה לפי סדר חשיבותן שמוגדר בעת יצירת סיפור משתמש).
7. כל מודול מייצג מערכת, הכוללת ממשק ניהול.

8. ניתן להוסיף רעיונות שלדעתכם מייעלים את המערכת (מודולים, משתמשים וכו').
9. דרישות המערכת יחולקו ל-3 ספרינטים, (ראו סילבוס הקורס+ תוכנית תרגולים).
10. יש לפתח את המערכת כך שתתמוך בלפחות שני תהליכים מקצה לקצה (E2E). הכוונה להגדרת תהליך הבנוי מרצף של פעולות המאפשרות לבצע משימה מורכבת.
11. פרויקטים המשלבים שפות פיתוח חדשות (כגון מטלב), חומרה ופתרונות שמחוץ לקופסא-יתבטא בתהליך הערכתם.

אפשר להשתמש במאגרים הנתונים הפתוחים של עיריית ב"ש open data smart7. להלן תיאור למעוניינים.

Open data smart7 – ניתן למצוא גם מאגרים בעיר אחרת...

בפרויקט יש לפתח מערכת web/mobile בגישת מידע פתוח בעיר חכמה תוך משתמש בפלטפורמת ה- Open Data Smart7. המטרה הכללית היא לשיפור איכות חייהם של התושבים באמצעות מידע הקיים בידי העירייה (לדגומה עיריית באר שבע). המאגר יכול להיות מועשר על ידי התושבים עצמם.

פיילוט Open data (מידע פתוח) בבאר שבע יצא לדרך לפני כשנתיים, במסגרת פרויקט Smart7 (עיר דיגיטלית), המהווה פיילוט ארצי בתמיכת מטה ישראל דיגיטלית במשרד לשוויון חברתי.

במסגרת הפיילוט, חשפה העירייה מאגרי מידע מגוונים בתחומים שונים הנוגעים לעיר לצפייה ולשימוש הציבור הרחב. כלל המאגרים מוגשים באמצעות פורטל ייעודי בפורמטים שונים, לנוחיותם של המשתמשים. כיום קיימים בו למעלה מ-32 מאגרים שונים, שעל חלקם אחראים גורמים נוספים מלבד העירייה, דוגמת משרד התחבורה, הלשכה המרכזית לסטטיסטיקה, תאגיד מי שבע, קהילת Tech7 ועוד.

לפרויקט 3 מטרות מרכזיות: הגברת השקיפות של העירייה; הפחתת עומס תפעולי על גורמי העירייה, הנגרם עקב בקשות פרטניות למסירת מידע; ובסיס לפיתוח חברתי-כלכלי מבוסס מידע (Data Driven Innovation), תוך חיזוק תרבות החדשנות בעיר ומשיכת גופים מחקר, פיתוח וכן גורמים עסקיים.

ניתן להתרשם מן הפורטל דרך הקישור הבא:

<https://www.beer-sheva.muni.il/OpenData/Pages/Data.aspx>

מוזמנים להצטרף לקהילת הפרויקט בפייסבוק: <https://www.facebook.com/groups/316432488804291>

בהיבט הטכני, ממשק הפעולות של המערכת הינו כלי עוצמתי למפתחים אשר פותח בפניכם אפשרות לתקשר עם האתר וכל המידע השמור בו.

ניתן למשוך את המידע מהפורטל בשתי דרכים:

1. הורדת הקובץ בפורמט המתאים – CSV\JSON\XML\SHP
2. להשתמש ב-API, על מנת למשוך את הנתונים בצורת בקשות GET

דוגמאות לשימוש ב-API – [כאן](#).

דוגמאות לקוד מוכן בכמה שפות ב-GitHub Repository – [כאן](#)

אפשר לבחון אפשרות לרכישת ביקון (מבוסס טכנולוגיית ה-Bluetooth) על ידי המחלקה-אנא הגישו בקשות במייל במהלך השבוע השני.

באופן כללי, ניתן להציב את החיישנים בחדר או בנקודה מסויימת במרחב הציבורי, ולשלוח מידע בדחיפה היישר לטלפון הנייד המתקרב לטווח השידור של הביקון, על פה המזהה ייחודי (ID) שלו. הטלפון שולח את המזהה לשרת, שבודק איזה פעולה יש להפעיל ל-ID הזה ולאחר מכן התשובה נשלחת חזרה למכשיר הפלאפון. עקב השימוש בפרוטוקול Bluetooth Low Energy (BLE) – הביקון איננו מצריך החלפת סוללה ולעיתים קרובות יכול להחזיק שנה (ואפילו 5 שנים) עם סוללה אחת.

שימושים אפשריים

מוסדות ציבוריים כמו מוסדות חינוך ומבני ציבור: ניתן למשל להתקין את הביקונים במכללות או בתי ספר, ולהנגיש מהם מידע לסטודנטים ותלמידים: שינויים במערכת השעות, חדשות על אירועים קרובים, שעות פתיחת הספרייה. כל אלו ישפרו את נוחות ויעילות התהליך הלימודי.

דוגמא נוספת- מוזיאון פארק קרסו למדע, שיכול להיעזר בביקונים בסמיכות למיצג הממותה המפורסם. כשהמבקרים יתקרבו למיצג, ניתן לשדר ישירות לפלאפונים שלהם סרטונים הממחישים את חיי הממותה לפני 11,000 שנים, מידע ביולוגי על משפחת הממותות, מפה אינטראקטיבית עם שטחי המחיה שלהן בעבר, והשמים הם הגבול!

שימוש ציבורי-קהילתי: אחת מיכולות הביקון היא ליקוט מידע מהמשתמשים הסובבים אותו. כך לדוגמא אנחנו יכולים לדעת (מבלי לאסוף מידע אישי על המשתמשים) כמה בעלי כלבים יש כרגע בגינת הכלבים והאם זה הזמן להוציא את הכלב שלי לשחק עם השאר?

להלן מדריכים לפיתוח באמצעות ביקונים, [לאנדראיד](#) וכן ל [iOS](#).

נעזר ב- [Estimote של Cloud](#), כדי להתחבר לביקונים ולהגדיר את אופן הפעולה שלהם. למידע נוסף על הטכנולוגיה:

<https://www.geektime.co.il/what-is-beacon-and-is-it-good-or-bad>

שיהיה בהצלחה!