**הנדסת תוכנה ואבטחת איכות תכנה במערכות מידע – מטלה 2**

Software Development Plan for “Hamka”



**החוג למערכות מידע אוניברסיטת חיפה**

**מגישים:**

|  |  |
| --- | --- |
| **מוחמד אדריס** | 314981598 |
| **אופיר סגל** | **205797210** |
| **שי בן חיים** | 308076553 |
| **ויאם שאהין** | 311182281 |

Contents

[1. היקף הפרויקט 3](#_Toc55923808)

[1.2. סקירה כללית – 3](#_Toc55923809)

[1.3. מטרת המסמך – 3](#_Toc55923810)

[2.. סימוכין 3](#_Toc55923811)

[3.סקירת העבודה הנדרשת 4](#_Toc55923812)

[3.1.דרישות ואילוצים על המערכת והתוכנה שיפותחו- אנו 4](#_Toc55923813)

[3.2.דרישות ואילוצים על תיעוד הפרויקט – 4](#_Toc55923814)

[3.3.מיקום הפרויקט במחזור חיי המערכת – 4](#_Toc55923815)

[3.4.תוכנה נבחרת ,אסטרטגיית רכישה או דרישות ואילוצים עליה – 4](#_Toc55923816)

[3.5.דרישות ואילוצים על לוחות זמנים ומשאבים של הפרויקט – 4](#_Toc55923817)

[3.6.אילוצים ודרישות אחרות- 4](#_Toc55923818)

[4.תכנון פעילויות לפיתוח התוכנה 4](#_Toc55923819)

[4.1.מודל פיתוח – 4](#_Toc55923820)

[4.2.תכניות כלליות לפיתוח 5](#_Toc55923821)

[4.2.1. שיטות פיתוח תוכנה – 5](#_Toc55923822)

[4.2.2. סטנדרטים- 5](#_Toc55923823)

[4.2.3. שימוש חוזר ברכיבי תוכנה 5](#_Toc55923824)

[5.תכנית מפורטת לביצוע פעילויות פיתוח תוכנה 5](#_Toc55923825)

[5.1.תכנון פרויקט ופיקוח 5](#_Toc55923826)

[5.1.1. תכנון פיתוח תוכנה- 5](#_Toc55923827)

[5.1.2. תכנון בדיקות מערכת- 6](#_Toc55923828)

[5.1.3. תכנון התקנות תוכנה- 6](#_Toc55923829)

[5.2.ניתוח דרישות מערכת 6](#_Toc55923830)

[5.2.1. ניתוח קלטי משתמש- 6](#_Toc55923831)

[5.3.מימוש תוכנה ובדיקות יחידה- 6](#_Toc55923832)

[5.4.הכנת מדריכים למשתמש- 7](#_Toc55923833)

[5.5.ניהול סיכונים, לרבות סיכוני ידועים ואסטרטגיות מתאימות- 7](#_Toc55923834)

# היקף הפרויקט

* 1. מערכת משחק המחשב "חמקה" תהווה חידוש למשחק הקופסא הקלאסי דמקה – ותכלול בנוסף שאלות טריוויה בנושאי הקורס בהנדסת תוכנה ואבטחת איכות, וכן אלמנטים סטטיסטיים ותחרותיים.
  2. סקירה כללית – הפרויקט עוסק בפיתוח משחק המחשב "חמקה" שמטרתו משחוק הלמידה של הסטודנטים בקורס הנדסת תוכנה ואבטחת איכות – משתמשיו העיקריים. הם יוכלו לרענן ולבדוק את הידע שלהם בקורס תוך כדי משחק בעזרת שאלות טריוויה המשפיעות על המשחק. שילוב אלמנטים תחרותיים יעלה את רמת המוטיבציה במהלך הסמסטר. הפרויקט יפותח ע"י קבוצת Chimps המונה 4 אנשי צוות. הפרויקט יימסר לצוות הקורס (הלקוח) ויעזור לו להשיג הבנה מעמיקה של רמת ההבנה בכיתה, בעזרת פאנל ניהול בו יוכלו לנהל את מאגר השאלות ולצפות במסך סטטיסטיקות. עד עתה הוגש מסמך דרישות מפורט בפורמט SRS – מסמך זה אושר ע"י הלקוח. למסמך הנוכחי מצורפים קובץ אב טיפוס שנעשה בתוכנת Eclipse ומסמך interfaces בו מוגדרת ארכיטקטורת המערכת.
  3. מטרת המסמך – במסגרת מסמך זה מפורטת תכנית פיתוח המערכת, תוך סקירת מודל פיתוח התוכנה, פעילויות מתוכננות, לוח זמנים, שיטות פיתוח, וסטנדרטים. בעקבותיו יחל הפיתוח בשלב האיטרציה הראשונה. להלן תיאור הפרקים הבאים במסמך:

|  |  |  |
| --- | --- | --- |
| מס' | שם הפרק | תיאור |
| 2 | סימוכין | רשימת המסמכים עליהם המסמך מתבסס |
| 3 | סקירת העבודה הנדרשת | סקירה כללית של שלבי העבודה |
| 4 | תכנון פעילויות לפיתוח התוכנה | פעילויות כלליות של תוכנית הפיתוח |
| 5 | תכנית מפורטת לביצוע פעילויות פיתוח תוכנה | פירוט שלבי העבודה הכוללים: תכנון, ניתוח, פיתוח, אינטגרציה, בדיקות, ניהול סיכונים והכנת מדריכים |
| 6 | לוחות זמנים | לוח זמנים ופעילויות של הפרויקט |

# סימוכין

* מפרט הדרישות נכתב בהתאם להמלצת ארגון IEEE Std 830-1998

בארגון לפי אובייקטים כמודגם בסעיף A.4 Template of SRS Section 3 organized by object למסמך.

[https://homepages.dcc.ufmg.br/~rodolfo/es-1-03/IEEE-Std-830-1998.pdf](https://homepages.dcc.ufmg.br/~rodolfo/es-1-03/IEEE-Std-830-1998.pdf%09)

* מפרט הדרישות נכתב בהסתמך על הנלמד בקורס "הנדסת תוכנה ואבטחת איכות 21 3610 א01".

# סקירת העבודה הנדרשת

* 1. דרישות ואילוצים על המערכת והתוכנה שיפותחו- אנו נפתח את המערכת בהתאם לדרישות שהוגדרו לנו בהוראות הפרויקט, ולפי מסמך הדרישות SRS שכתבנו במטלה הקודמת. שפת כתיבת המערכת תיהיה JAVA ולפי ארכיטקטורת MVC.
  2. דרישות ואילוצים על תיעוד הפרויקט – יתועדו בדוחות כל שלבי חייו של הפרויקט, בין השאר שלבי הניתוח, הפיתוח, וההטמעה. ייעשה תיעוד של ההתקדמות בפרויקט, וחלוקת המשימות באמצעות גיליון Excel רכיבי הקוד יהיו מתועדים בעזרת Javadoc והערות נוספות בגוף הקוד.
  3. מיקום הפרויקט במחזור חיי המערכת – הפרויקט נמצא כעת בשלבי הניתוח והעיצוב של המערכת. המסמך הוא קדם לשלב הפיתוח.
  4. תוכנה נבחרת ,אסטרטגיית רכישה או דרישות ואילוצים עליה – הפיתוח ייעשה בשיטת ה- Agile בה משתמשים ב-Strategy Incremental .הפיתוח יורכב מסדרת שחרורי builds )גרסאות עובדות(. כל גרסה תרחיב פיצ'רים שכבר מומשו, תוסיף חדשים, ותיבדק ע"י המפתחים והן מול הלקוח )במפגשים שייקבעו עמו במהלך הפיתוח(. ה-build האחרון יכלול מימוש מלא של הדרישות.
  5. דרישות ואילוצים על לוחות זמנים ומשאבים של הפרויקט – לוח הזמנים לפיתוח המערכת כפוף ללוח הזמנים של הקורס במהלך הסמסטר. זמן הפיתוח מוגבל מאוד (כחודשיים), ועל כן יש לממש את הדרישות החשובות ביותר בהקדם.

צוות הפרויקט כולל 4 אנשי צוות בתפקידים הבאים:

**Product Manager** – דואג לרצונות הלקוח ולעמידה בדרישות, אחראי על המוצר הסופי.

**Scrum Master** – מנהל רשימת הדרישות, פיצול לאיטרציות, חלוקה של אנשי הצוות למשימות, מעקב ועזרה לאנשי הצוות.

**מפתח** – אחראי על פיתוח מחלקות התונים (בדגש על שכבת ה-Model), קריאה וכתיבה מקבצי ה-JSON, פיתוח חלק ממחלקות ה-Controller.

**איש UI** – אחראי על פיתוח ממשק המשתמש, מסכים ותפריטים. אחראי על חווית המשתמש ונוחיות השימוש במערכת. יהיה חלק מפיתוח מחלקות ה-Controller.

* 1. אילוצים ודרישות אחרות- הצוות יפתח את המערכת הממוחשבת תוך מפגשים משותפים תכופים.

1. תכנון פעילויות לפיתוח התוכנה
   1. מודל פיתוח – המודל הנבחר הוא גישת Agile המאופיינת בעבודה באיטרציות תכופות. כל איטרציה מחולקת לשלבים, וכל שלב מפותח בצורה מהירה, "ספרינט". בכל שלב יש לנו תוכנה עובדת עם רכיב נוסף עובד. לכן גישה זו מתאימה לפיתוח המשחק, ולדרישות הלקוח, הכוללות בדיקה התקדמות אחת לתקופה בזמן הפיתוח. נעדכן את ההתקדמות שלנו במשימות בקובץ Excel עפ"י דרישות הלקוח, וכן באפליקציית Trello בה ננהל גם את הטיפול בבאגים. ננהל את הגרסאות בעזרת תוסף SVN.

## תכניות כלליות לפיתוח

* + 1. שיטות פיתוח תוכנה – פיתוח המשחק יבוצע עפ"י מודל הפיתוח האג'ילי המאופיין באיטרציות רבות. המשחק ייכתב בשפת התכנות Java תוך שימוש בספריית JavaFX. ארכיטקטורת התוכנה תהה מבוססת על תבנית ה-MVC. נשתמש בגישת TDD ונפתח את קוד הבדיקה לפני קוד המערכת עצמה
    2. סטנדרטים- הסטנדרטים יוגדרו ע"י הלקוח (צוות הקורס). בנוסף קיימים סטנדרטים במסמך הדרישות שהוגש כמטלה 1 שבהם נצטרך לעמוד במהלך הפיתוח. בנוסף יש סטנדרטים לכתיבת קוד (יעיל, מתועד וכד').

### שימוש חוזר ברכיבי תוכנה

* + - 1. שימוש חוזר ברכיבים קיימים- הספריות והרכיבים שנשתמש בהם לכתיבת המערכת יוכלו לשמש לבנייה של מערכות דומות בעתיד. ניתן יהיה להשתמש במסכים ובפונקציונאליות לפרויקטים אחרים הבנויים על פי שיטת MVC.
      2. פיתוח רכיבים לשימוש חוזר- לא יתבצע פיתוח רכיבים לשימוש חוזר במערכת.

1. תכנית מפורטת לביצוע פעילויות פיתוח תוכנה: במהלך הפרויקט נעבוד ב-3 איטרציות ע"פ התכנית שקיבלנו מצוות הקורס. בכל איטרציה נבנה חלק נוסף של המערכת ובסופן נגיש אב טיפוס של המשחק הממוחשב "מונופול" שיכיל את כל הפונקציונליות הדרושה.

## תכנון פרויקט ופיקוח

* + 1. תכנון פיתוח תוכנה- פירוט האיטרציות להלן:

|  |  |
| --- | --- |
| **איטרציה 1** | Implementing the basic logit of the game |
| משימה 1 | הגדרת מבנה MVC וסיווג המחלקות (אלה שכבר כתבנו במסמך ה-SRS) לשכבות השונות תוך התחשבות בכללי ה-MVC. |
| משימה 2 | הגדרת ומימוש מחלקת SysData שתהיה בשכבת ה-Model. דגש על קריאה, עדכון וכתיבת הנתונים מקבצי ה-JSON |
| משימה 3 | מימוש חלקי של התפריט הראשי של המערכת Menu השייך לשכבת ה-View |
| משימה 4 | ביצוע בדיקות עבור החלקים שמימשנו עד כה |
| **איטרציה 2** | Implementing the GUI of the game |
| משימה 1 | בניית מחלקות שכבת ה-Controller המקשרות בין שכבת הView- לשכבת ה-Model |
| משימה 2 | מימוש חלקי של מסך לוח המשחק board ,כולל הצגת התור של כל שחקן |
| משימה 3 |  |
| משימה 4 |  |
| **איטרציה 3** | Building a more full-featured version |
| משימה 1 | מימוש חלונות ה-popUp המציגים את השאלות וקולטים את תשובת השחקן |
| משימה 2 | מימוש חלונות ה-popUp האחראיים על הודעות החיווי (למשל: כששחקן שגה בשאלה תוצג לו הודעת חיווי עם התשובה הנכונה(. |
| משימה 3 | מימוש מסך הצגת ההיסטוריה GamesHistory השייך לשכבת ה-view. |
| משימה 4 | ביצוע בדיקות נוספות על חלקי הקוד שמימשנו באיטרציה זו. |
| משימה 5 | סיום מימוש כל מסכי ופונקציונאליות המשחק. |

פירוט איטרציה 1:

|  |  |  |  |
| --- | --- | --- | --- |
| **איטרציה** | **דרישה** | **משימה** | **מבצע** |
| **איטרציה 1-**  Implementing the basic logic of the game | פתיחה תפריט ראשי | מימוש תפריט ראשי Menu של המשחק ללא פונקציונליות, ללא מעבר למסכים שונים בלחיצות על כפתורים | Frontend Developer |
| מימוש מחלקת SysData כתיבת תכונות, setters, getters. | Backend Developer |
| כתיבת מתודות קריאה, כתיבה ועדכון מקבצי ה-JSON. | Backend Developer |
| כתיבת Acceptance Test | PO |
| התחלת משחק | מימוש מחלקת Game: כתיבת תכונות,getters, setters. | Backend Developer |
| מימוש שאר המחלקות  Board, Tile, yellowTile וכו' (מפורט בטבלה מעלה) | Frontend Developer |
| כתיבת Acceptance Test | PO |
| ניהול מאגר שאלות | מימוש מחלקת questions כתיבת תכונות, setters, getters. | SM |
| שמירת היסטורית משחקים | מימוש מחלקת players כתיבת תכונות, setters, getters. | SM |
| מימוש מחלקת points כתיבת תכונות, setters, getters. | PO |

* + 1. תכנון בדיקות מערכת- הבדיקות יעשו לפי מודל TDD (פיתוח מונחה בדיקות), כלומר הבדיקות ייכתבו לפני כתיבת התוכנה.
    2. תכנון התקנות תוכנה- פתיחת המשחק תתבצע ע"י לחיצה כפולה בעכבר על קישור התכנה ומעבר לקובץ JAR או EXE.

## ניתוח דרישות מערכת

* + 1. ניתוח קלטי משתמש- במהלך המשחק המשתתפים ייעזרו בעכבר ומקלדת שיהוו את אמצעי הקלט.
  1. מימוש תוכנה ובדיקות יחידה- בדיקות יחידה יבוצעו ע"י Junit ,שזהו כלי פתוח וחופשי לבדיקות יחידה בשפת התכנות Java .את הבדיקות נריץ לאחר כתיבת כל קטע קוד אליו נדרשת בדיקה במסמך הבדיקות TDD)) במידה וקיימת בעיה- נתקן אותה מיידית לפני המשך הכתיבה.

|  |  |  |
| --- | --- | --- |
| תוצאה מתוכננת | תיאור | # |
| המתודה checkAns, בודקת את תשובתו של המשתמש אל מול התשובה הקשורה לאותה שאלה במבנה הנתונים של השאלות והתשובות. במידה והתשובה נכונה יוחזר true ויופיע מסר למשתמש עם הכיתוב "תשובה נכונה" במידה ויוחזר false יוצג המסר "תשובה שגויה, התשובה הנכונה היא \_\_" | בדיקת התשובה שהזין המשתמש בהופעת השאלה שהוצגה למשתמש במהלך המשחק. | 1 |
| המתודה loadQuestions קוראת את קבצי ה-JSON במהלך הפעלת המערכת או במהלך עדכון השאלות בקובץ. במידה ולא ניתן לקרוא או לכתוב את הנתונים יוצג מסר למשתמש "הנתונים לא נקלטו כראוי" | בדיקה שקבצי ה-JSON המכילים את השאלות במשחק מתעדכנים במערכת. | 2 |
| במתודה addPlayer תתבצע בדיקה לראות אם שם המשתמש שהזין המשתמש איננו תפוס כבר ע"י משתמש אחר, במידה והשם שהזין המשתמש לא זמין תופיע ההודעה "שם המשתמש שהזנת שייך כבר למשתמש אחר". במידה והשם תקין יוחזר true והרישום יעבור בהצלחה. | בדיקת שם המשתמש של השחקן מול מאגר השחקנים במערכת | 3 |
| המתודה checkTimer תבדוק שבמהלך כל תור של המשתמש, הטיימר הרלוונטי לאותו תור מתאפס. במידה והתור מתאפס בהצלחה יוחזר true והמשחק ימשיך כרגיל, במידה והוא לא התאפס יוחזר false ותופיע הודעת השגיאה "טיימר לא התאפס" | בדיקת איפוס שעון הטיימר בחידוש תורו של המתשמש | 4 |
| המתודה checkWinner תבדוק את מספר הנקודות שכל משתמש צבר לאורך המשחק ותחזיר את קוד השחקן שניצח ותוצג הודעה "השחקן \_\_\_ ניצח!", במידה ויוחזר ערך 0 המערכת תשלח מסר "המשחק נגמר בתיקו" | בדיקת ניצחון של המשתמש. משתמש שניצח במשחק הוא השחקן בעל מס' הנקודות המקסימלי לאורך המשחק. | 5 |

* 1. הכנת מדריכים למשתמש- למשחק יצורפו מסך הוראות לשימוש במשחק. בנוסף, בממשק המשתמש נוסיף כפתור של עזרה שבו יהיה פירוט של כל האפשרויות שאפשר לבצע בכל מסך.
  2. ניהול סיכונים, לרבות סיכוני ידועים ואסטרטגיות מתאימות- קיים סיכון למחיקת קוד הכרחי לתוכנה ע"י טעות אנוש או משתמש לא מיומן, נעשה מאמצים למנוע זאת ע"י גיבוי הקוד ושמירת גרסה נוכחית וקודמת (שמירת היסטוריה) ועדכון הצוות בכל שינוי שמתבצע, בנוסף נוסיף תיאורים בטקסט לפני כל קטע בקוד שיוסבר מה קטע זה עושה קיים סיכון לנפילת התכנה, ננסה לטפל בסיכון זה ע"י גיבוי הנתונים והתאוששות מהירה. קיים סיכון לחשיפה לפגיעה מכוונת כמו וירוסים, ננסה למנוע זאת ע"י אנטי וירוס שידאג למניעת וירוסים, אבחונם ואחזור (טיפול בהם) .קיים סיכון שמישהו מחברי הצוות לא יוכל למלא את חלקו (כתוצאה מסיבות פרטיות למשל), מה שיכול לפגוע בעמידה בלוחות הזמנים לרבות זמן הגשת הפרויקט. במקרה כזה שאר חברי הצוות ישתדלו למלא את מקומו ולהגביר את קצב העבודה.

# לוחות זמנים

|  |  |  |
| --- | --- | --- |
| **#** | **תיאור משימה** | **תאריך ביצוע** |
| 1 | SRS | 01.11.2020 |
| 2 | הגדרת SDP ואיטרציות | 15.11.2020 |
| 3 | בניית אב טיפוס וקבלת משוב לקוח | 16.11.2020 |
| 4 | איטרציה 1 | 29.11.2020 |
| 5 | איטרציה 2 | 13.12.2020 |
| 6 | איטרציה 3 | 27.12.2020 |
| 7 | סגירת פרוייקט | 18.1.2021 |
| 8 | תרגיל Swap | 21.1.2021 |

