לכבוד

24/4/2022

יחידת הפרויקטים

מה"ט

**הצעה לפרויקט גמר**

**1. פרטי הסטודנטים:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **תאריך סיום הלימודים** | **טלפון נייד** | **כתובת** | **ת.ז .9 ספרות** | **שם הסטודנט** |
| 5 לספטמבר 2019 | 058-7299721 | חזון איש 4, זכרון יעקב | 322469644 | עדי בוסתן |
| 5 לספטמבר 2019 | 054-2680100 | האפיק 333, בארותיים | 207882176 | בר זסלבסקי |

שם המכללה: בסמ"ח

סמל המכללה: 71605

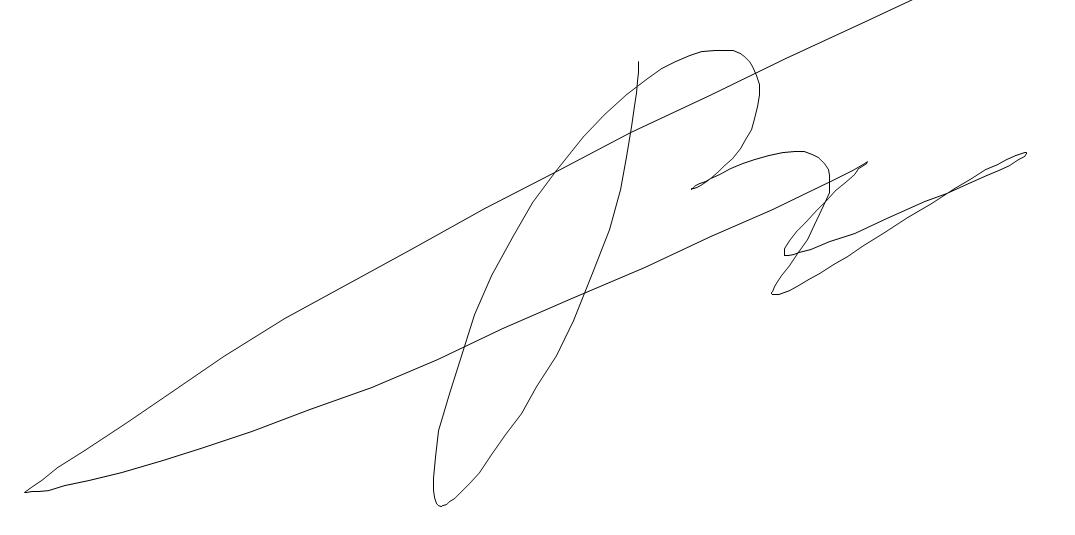
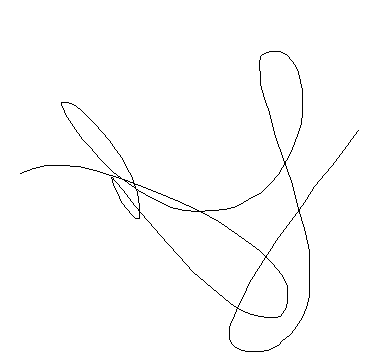
מסלול הכשרה: טכנאים מוסמכים

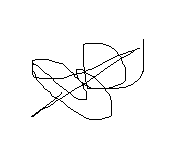
מגמת לימוד: הנדסת תוכנה

מקום ביצוע הפרוייקט: בסמ"ח

**2. פרטי המנחה האישי:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **מקום עבודה / תפקיד** | **תואר** | **טלפון נייד** | **כתובת** | **שם המנחה** |
| מתכנת בחיל האוויר | תואר ראשון: הנדסת תוכנה  תואר שני: מדעי המחשב | 0546616220 | דרך יתיר 98, מיתר | נדב סקלר |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
 חתימת סטודנט חתימת סטודנט**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **חתימת הגורם המקצועי מטעם מה"ט חתימת המנחה האישי**

1. שם הפרויקט  
   Own My Files
2. רקע
   1. **תיאור ורקע כללי**כיום ביחידתנו ישנה תיקייה רשתית המשותפת לכולם בה נמצאים הרבה קבצי טקסט אישיים של אנשים שונים ביחידה אותם לא ירצו שכלל אנשי היחידה יוכלו לראות. אומנם הקבצים מסודרים לכל אחד בתיקייה אישית, אך הדבר לא מונע גישה לקבצים אלה. בשביל להבטיח את פרטיותו של כל אדם חשבנו על פיתוח מערכת אשר תצפין ותפענח את הקבצים של כל אחד בצורה קלה ונוחה למשתמש.
   2. **מטרות המערכת**מטרת המערכת היא להצפין ולפענח קבצים ותיקיות בעלות מידע שלא רוצים שיהיה גלוי לכולם ויהיה גלוי אך ורק לאנשים בעלי מפתח מתאים לקובץ שהם רוצים לפתוח.
3. סקירת מצב קיים בשוק, אילו בעיות קיימות  
    **כיום, המערכות אשר מנגישות את היכולת הזו הן לא אינטואיטיביות למשתמש.**  
   **בנוסף, בעת הצפנה של מספר רב של קבצים או תיקיות, ההצפנה לוקחת יותר מדי זמן.**
4. מה הפרויקט אמור לחדש או לשפר  
   **הפרויקט מנגיש את פעולות ההצפנה והפיענוח ללקוח בצורה נוחה וקלה לתפעול מההצעות שנמצאות היום בשוק. בנוסף, כאשר תהיה דרישה להצפין מספר גדול של קבצים, המערכת בעלת ביצועים גבוהים – מה שאמור לשפר את מהירות ההצפנה/פענוח.**
5. דרישות מערכת ופונקציונאליות
   1. דרישות מערכת  
      5.1.1. **על המערכת לעבוד במהירות גבוהה בעת עומסים – גם כאשר משתמשים רבים פונים למערכת וגם כאשר משתמש יחיד מנסה לפענח או להצפין מספר רב של קבצים.**5.1.2. **על המערכת לספק אבטחה מקסימלית כדי לשמור על המידע החסוי – כך שאנשים שאין להם גישה למפתח לא יוכלו לראותו.**
   2. דרישות פונקציונאליות  
      5.2.1. **הדרישה הבסיסית של המערכת היא לפענח קובץ יחיד עם מפתח שיקבל לאחר ביצוע הצפנה.**5.2.2. **המערכת תהיה מאובטחת ע"י שם משתמש וסיסמא אותם יצטרך המשתמש להזין בעת חיבור למערכת.**5.2.3. **המשתמש יוכל להצפין ולפענח תיקייה שלמה בעזרת אותו המפתח.**5.2.4. **המשתמש יוכל לבחור בין חמישה סוגים שונים של הצפנות ובהתאם חמישה סוגים שונים של פענוח.**5.2.5. **המשתמש יוכל לבחור איך לבצע את ההצפנות – קובץ קובץ, או את כל הקבצים בו-זמנית.**5.2.6. **המשתמש יוכל לשמור את סוגי וסדר ההצפנה בקובץ JSON וליבא אותו לאחר מכן.**
6. בעיות צפויות במהלך הפיתוח ופתרונות
   1. תיאור הבעיות  
      6.1.1. **המערכת לא תצליח להתמודד עם מספר רב של קבצים אשר המשתמש ירצה להצפין בו-זמנית.**6.1.2. **איבוד המפתח בעת ההצפנה או החלפת המפתח בין הצפנות שונות.**6.1.3. **בעבודה על מספר קבצים בו-זמנית, בעיה בסנכרון העבודה על הקבצים – יכול לגרום להצפנה כפולה על אותו הקובץ או על אי הצפנה כלל של קובץ.**
   2. פתרונות אפשריים  
      6.2.1. **דאגה לשרת חזק שיעמוד בעומסים.**6.2.2. **כתיבת קוד נקי והפרדה בין תפקידי מחלקות וביצועים.**6.2.3. **כתיבת קוד יעיל ומסודר תוך שילוב עבודה עם נימים.**
7. פתרון טכנולוגי נבחר
   1. טופולוגית הפתרון  
      **המערכת מורכבת מצד שרת האחראי על הצפנה ופענוח של הקבצים, צד לקוח WEB המתקשר עם צד השרת ו-DB אשר אחראי לנתוני ההתחברות וההרשאות. כפי שניתן לראות בתרשים:**Diagram

      Description automatically generated
   2. טכנולוגיות בשימוש  
      **בפיתוח צד הלקוח נעזר בספריית RXJS אשר מציעה תכנות א-סנכרוני עם observable streams .  
      ספריה זו בעצם נותנת יכולת של התבוננות והבחנה לקוד – כך שהקוד מחכה שיקרה משהו ולאחר שמשהו קרה (לדוגמא, לחיצת כפתור) הקוד יבצע פעולה בהתאם.  
      את צד השרת נפתח באמצעות JAVA וספריית spring boot אשר מנגישה לנו באופן נוח וקל בקשותhttp בין הלקוח לשרת.**
   3. שפות הפיתוח  
      **לצד השרת נשתמש בשפת JAVA – זאת מפני כי זוהי שפה נוחה לעבודה עם נימים ושפה מונחת עצמים – דבר שמקל עלינו לשמור על סדר ו**- **clean Code בכדי לשמור על המפתח במקום נפרד.  
      לצד הלקוח נשתמש ב**- **Angular – שפה נוחה וקלה לשימוש ובעלת הרבה דוקומנטציה לעזרה.  
      בנוסף, שפה זו בעלת אופי מודולארי, משמע שניתן לחלק את הקוד בצורה יעילה למודלים. זה מאפשר לארגן את הפונקציונליות של המערכת בצורה קלה.**
   4. תיאור הארכיטקטורה הנבחרת  
      **פרטי המשתמשים במערכת ישמרו מיד בעת רישומם ב- DB. כאשר המשתמש יכנס אל היוזר שלו, יעמדו בפניו שתי אפשרויות.**

א.הצפנת קובץ **- על המשמש יהיה לבחור את הקובץ אותו ירצה להצפין.**

**לאחר מכן יהיה עליו לבחור את סוג ההצפנה אותו ירצה לבצע. המערכת "תגריל" מספר רנדומלי אשר יהווה למפתח של אותו קובץ. לבסוף, הקובץ המוצפן ישמר באותו נתיב ממנו הקובץ המקורי נלקח. שם הקובץ, סוג ההצפנה, והמפתח, ישמרו בטבלה ב- DB אליה ניגש בעת ביצוע פיענוח.**

ב.פענוח קובץ **- על המשתמש יהיה לבחור את הקובץ אותו ירצה לפענח. בקשה זו תשלח אל ה- DB ותשלוף משם את המפתח וסוג ההצפנה בהתאם לשם הקובץ אותו יכניס. המערכת תפענח את הקובץ ותשמור אותו בנתיב המקורי.**

* 1. חלוקה לתכניות ומודולים  
     **נחלק את ארכיטקטורת המערכת לשני חלקים. בצד הלקוח, הארכיטקטורה תהיה מורכבת על בסיס לוגיקה של קומפוננטות, ולכל קומפוננטה יהיו מרכיבים משלה. בנוסף, יהיו מחלקות משותפות של רכיבים שכמה קומפוננטות משתמשות בהם.   
     בצד השרת תהיה חלוקה לפי סוגי הצפנות ויכולות – לכל סוג הצפנה יהיה מודול נפרד ודברים בסיסיים המשותפים לכלל ההצפנות יהיו גם כן בנפרד.**
  2. סביבת השרת  
     **השרת יהיה מכונה וירטואלית המריצה לינוקס – את צד השרת נסגור כ RPM ונריץ על השרת.**
  3. ממשק המשתמש/ לקוח- GUI  
     **ממשק המשתמש ירוץ ב OpenShift – דבר המאפשר לנו שרידות גבוהה של המערכת ועמידה בעומסים.**
  4. ממשקים למערכות אחרות\ API  
     **למערכת אין התממשקות למערכות אחרות.**
  5. שימוש בחבילות תוכנה  
     **rxjs – לשימוש בצד הלקוח, spring boot – API בין צד השרת לצד הלקוח וספריית IO לכתיבה וקריאה מקבצים בצד השרת.**

1. שימוש במבני נתונים וארגון קבצים  
   **נשמור את המידע החשוב לנו (המידע על המשתמשים ומפתחותיהם) בMongo DB,** **מפני ששומר נתונים בצורת JSON (צורה שנוחה לנו לשימוש) ויעזור לנו לביצוע שליפות מהירות ולא מורכבות.  
   את קבצי ההצפנה והפענוח נשמור מקומית על המחשב האישי של כל משתמש (במיקום לפי בקשת המשתמש).**
2. תרשימי מערכת מרכזיים
   1. Use Case:  
      **כל משתמש הרשום במערכת יהיה רשאי להצפין ולפענח איזה קובץ שירצה בעזרת המפתחות השייכים אליו.  
      בנוסף, תיתנן לו האפשרות לבחור בין חמשת צורות ההצפנה והפענוח – כאשר בסוף תהליך ההצפנה יקבל המשתמש מפתח חדש אותו יוכל לשתף.**
   2. Diagram

      Description automatically generatedSequence diagram:
   3. Diagram

      Description automatically generatedData flow:
3. תיאור/ התייחסות לנושאי אבטחת מידע  
   **ישנה התייחסות קריטית אחת לנושא אבטחת המידע והיא שהמפתח של כל משתמש לא ידלוף/יגיע למשתמש אחר – את בעיה זו נפתור בעזרת כך שלכל סוג משתמש יהיה token שמור בשדה ב- DB וכך נוכל לאמת את זהותו בכל פעם ולראות מהן הפעולות אותן הוא מורשה לבצע. בעת כניסה ראשונית למערכת, המשתמש יכנס ובעזרת הזדהות של שם משתמש וסיסמא נקבל את ה- token הספציפי שלו, הוא יישמר   
   ב- local storage וכך לא יצטרך להזדהות בכל פעם מחדש.**
4. משאבים הנדרשים לפרויקט
   1. ציוד נדרש  
      **הפיתוח יעשה על גבי מחשבי ווינדוס ובעלי גישה לשרת הווירטואלי של לינוקס. בנוסף, שרת לינוקס וירטואלי ומשאבים בOpenShift.**
   2. תוכנות נדרשות  
      **על מנת לעבוד עם ג'אווה, נזדקק לתוכנת IntelliJ. ועל מנת לעבוד עם Angular נזדקק לתוכנת WebStorm. זאת בנוסף להתקנה של Angular, ג'אווה וspring boot. תוכנת MobaxStream לעבודה על שרת לינוקס וירטואלי.**
   3. ידע חדש שנדרש ללמוד לצורך ביצוע הפרויקט  
      **לצרוך ביצוע הפרויקט נדרש מאיתנו ללמוד angular לצד הלקוח ועבודה עם API של mongo DB.**
   4. ספרות ומקורות מידע  
      **דוקומנטציה של הספריות והשפות תכנות, ניסיון של מלווה הפרויקט, וstack overflow.**
5. תוכנית עבודה ושלבים למימוש הפרויקט:  
   **נקדיש כ120 שעות ללמידה ופיתוח צד הלקוח, 180 שעות לצד השרת מפני ששם יושבת רוב מורכבות האלגוריתמיקה, כ10 שעות לעבודה עם הDB ועוד כ20 שעות לבדיקות וCI-CD.**
6. תכנון הבדיקות שיבוצעו

|  |  |
| --- | --- |
| **רישום משתמש חדש** | **הוספת ב המשתמש לDB בדיקת יצירת הtoken.** |
| **התחברות עם פרטים שגויים** | **הצגת הודעת שגיאה מתאימה.** |
| **התחברות עם פרטים נכונים** | **העברה לדף הבית של המשתמש.** |
| **ביצוע הצפנה על קובץ** | **קבלת קובץ מוצפן ומפתח בהתאם.** |
| **פענוח קובץ בדרך שגויה** | **קבלת קובץ מוצפן.** |
| **פענוח קובץ בדך נכונה** | **קבלת קובץ זהה למקורי.** |

1. בקרת גרסאות (version control)  
   **ביצוע בקרת התצורה יתבצע באמצעות GitHub.**
2. **הערות ראש המגמה במכללה**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **אישור ראש המגמה**

שם: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ חתימה: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ תאריך: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **הערות הגורם המקצועי מטעם מה"ט**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **אישור הגורם המקצועי מטעם מה"ט**

שם: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ חתימה: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ תאריך: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_