

**פרוייקט מעשי לתואר B.Sc. במדעים**

**אפליקציה לשמירת סיסמאות ומידע מוצפן**

**Secure Vault**

**מגישים:** אורי אבן חן יבגני איסייב

עידן ברדוגו יותם שובל

**בהנחיית:** רועי זימון **חתימה:**

**תאריך הגשה:** 01.12.2019

**רקע:**

ככל שהטכנולוגיה מתקדמת כך גוברת הדרישה לפרטיות מצד המשתמשים.

ריבוי סיסמאות לחשבונות דיגיטליים עוזרת לנו להגדיל את הפוטנציאל לאבטחת המידע האישי שלנו ברשת. עם זאת הקושי לזכור את הסיסמאות מביא לשמירתן בצורה לא מאובטחת על גבי המכשיר הסלולרי או באופן פרימיטיבי.

אובדן המכשיר הסלולרי או מציאת הסיסמאות ע"י גורם עוין עלולים להביא במקרה הקל לפגיעה בפרטיות ובמקרה המורכב לפגיעה נפשית, כלכלית, תדמיתית הלכה למעשה.

כיום, קיימים מעט מאד פתרונות רווחים לאדם הממוצע.

הפרויקט יקדם פתרון לאחסון הסיסמאות והמידע בצורה מאובטחת תוך לקיחה בחשבון של כל אילוצי אבטחת המידע וחווית השימוש של הלקוח.

**מטרת הפרויקט:**

האפליקציה תאפשר שמירת סיסמאות ומידע בצורה מאובטחת.

האפליקציה תאפשר יצירת שני שלבי אימות למשתמש.

הממשק יאפשר בחירת רמת אבטחה בהתאם לעדיפות המשתמש ובהתאמה להרגלי השימוש שלו.

**מקרא מונחים:**

**רמות אבטחה (Security levels)** - קיימות 3 רמות אבטחה אשר מגדירות את מאפייני השימוש באפליקציה, כגון: אופן הצגת המידע, שלבי אימות ועוד.

**רשומה (Record)** - הישות המרכזית באפליקציה המכילה את כל הסיסמה והמידע הרלוונטיים לאתר אינטרנט, רשת חברתית, חשבון בנק ועוד.

**סוג רשומה (Type)** - לכל סוג רשומה מוגדרים שדות אחרים המותאמים למאפייני הרשומה. לדוגמא: ברשומה מסוג כרטיס אשראי יופיע מספר cvv מה שלא יופיע ברשומה מסוג אתר אינטרנט.

**קטגוריה/תיקייה (Category\Folder)** - אקטיביטי המציג את כל הרשומות המוגדרות מסוג זה.

**דרישות הפרויקט:** דרישות לא מנדטוריות בצבע שונה

**דרישות פונקציונליות Functional Requirements:**

* כניסה למערכת - תתאפשר ע"י הכנסת שם משתמש וסיסמת מאסטר.
* המשתמש יוכל לבחור כניסה באמצעות מספר שלבים (סיסמה, ציור, טביעת אצבע) בהתאם ליכולות המכשיר.
* הצגת כל הרשומות עפ"י העקרונות הבאים:
* כל הרשומות יהיו מסודרות בקטגוריות.
* המשתמש יוכל לבחור בעצמו את שמות הקטגוריות.
* ניתן לבצע חיפוש ידני אשר יציג את הרשומות בהן המחרוזת מופיעה בכותרת הרשומה.
* כל רשומה תוצג עפ"י העקרונות הבאים:
* רק אחרי לחיצה על סמל ייעודי (עין) הסיסמה המקורית תופיע.
* יהיה ניתן להיכנס לאתר אינטרנט ע"י לחיצה על קישור.
* הזנת רשומה חדשה מהסוגים הבאים:
* אתר אינטרנט ואימייל.
* כרטיסי אשראי.
* פספורט/ת.ז.
* פתק (note).
* חשבון בנק.
* רשתות חברתיות.
* רכישות אונליין.
* רישיון נהיגה.
* עקרונות:
* תוצג אינדיקציה על חוזק הסיסמה.
* כל הרשומות ישמרו במכשיר הפלאפון בהצפנה.
* בכל רשומה יהיה שדה הערה שבו המשתמש יוכל להוסיף מידע לפי צרכיו.
* עריכת רשומה תתאפשר לאחר הכנסת סיסמת תבנית (בהתאם לרמת אבטחה).
* המשתמש יוכל להגדיר תוקף לסיסמה. האפליקציה תזכיר לו לשנות את הסיסמה.
* האפליקציה תגדיר רמת אבטחה מומלצת כברירת מחדל.
* ביציאה מהאפליקציה למשך יותר מעשר דקות תידרש כניסה מחדש.

**דרישות לא פונקציונליות Non Functional Requirements:**

* האפליקציה תכתב בשפת JAVA
* מידע ונתונים הנשמרים במערכת יוכנסו לבסיס הנתונים כאשר הם מוצפנים.
* ממשק נוח להכנסת נתונים אינטואיטיבי.
* רמת אבטחה הניתנת לבחירה בהתאם לרגישות המידע.
* האפליקציה תרוץ מהר ותגיב מהר למשתמש.
* האפליקציה תסייע למשתמש בניהול מאגר הסיסמאות והמידע האישי.
* האפליקציה תאחסן את המידע על מכשיר באופן לוקאלי.

ניתוח איומים:

* חשיפה פרטי משתמש - מתקפת MITM.
* חשיפת פרטי מסד הנתונים.
* נוזקה על מכשיר המשתמש - קילוגר \ דלת אחורית.
* גניבת המכשיר של המשתמש.
* שינוי הגדרות ושליפת מידע - תוקף ניגש פיזית אל המכשיר.

בדיקות יחידה TTD:

בדיקות UI - בבדיקות ממשק משתמש השתמשנו בספריית Espresso. הספרייה נותנת שלל אפשרויות - ביצוע בדיקות שמדמות משתמש שלוחץ על כפתורים, רושם משתמש חדש, מכין רשומה חדשה, מכניס טקסטים, מכניס pattern, לוחץ חזור, משתמש בחיפוש.

ראשית יש לעבור על הטסטים של ה- MainActivity, לעבור על הטסט שנקראtestNewUserRegister ולאחר מכן ניתן לעבור על שאר הטסטים של MainActivity.

כשבודקים את RecordRecycler יש להריץ את isItemCreated על מנת ל"אכלס" את הRecycler בכמה רשומות עליהן יהיה ניתן לבצע את שאר הבדיקות.

\*הערה - לעיתים ה- Pattern לא עובד ויש לנסות להריץ שוב את הטסט.

* MainActivityTest
* <https://youtu.be/lae-VeC_U14>
* testNewUserRegister - מבצעת רישום משתמש חדש
* testSignIn - מבצעת כניסת משתמש רשום
* TestValidPassword - בודק ולידציית סיסמאות
* TestValidEmail - בודק ולידציית אימייל
* CategoryList\_ActivityTest
* <https://youtu.be/Wz_grlakhpc>
* CategoryClicked - בודק שהקטגוריה הנכונה נפתחת
* SearchClicked - בודק שהרשומה הנכונה מתקבלת בחיפוש
* newUser\_ActivityTest
* <https://youtu.be/9rYXj7sjQHY>
* TestValidUsername - בודק ולידציית שם משתמש
* TestValidPassword - בודק ולידציית סיסמאות
* TestVerifyPassword - בודק האם הסיסמא דומה לוידוא סיסמא
* TestValidEmail - בודק ולידציית אימייל
* RecordRecycler\_ActivityTest
* <https://youtu.be/TrWC3oHKn9A>
* isItemCreated - בודקת אם נוצרו רשומות חדשות.
* isItemAtPositionShowed - בודקת אם נפתחת הרשומה הרלוונטית.
* isItemDeleted - בודקת אם הרשומה שבחרנו נמחקת.
* isItemUpdated - בודקת אם הרשומה שבחרנו מתעדכנת.

תרשים מחלקה - Class Diagram:



מודל ישויות קשרים - ERD:

