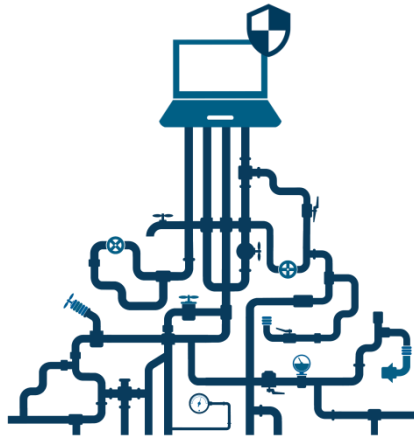


המחלקה להנדסת תוכנה

לוח קונפיגורציות בטכנולוגיות אינטרנט

Web Configurations Dashboard



חיבור זה מהווה חלק מהדרישות לקבלת

תואר ראשון בהנדסה

מאת

אלון שוורץ ואלכס לוינזון

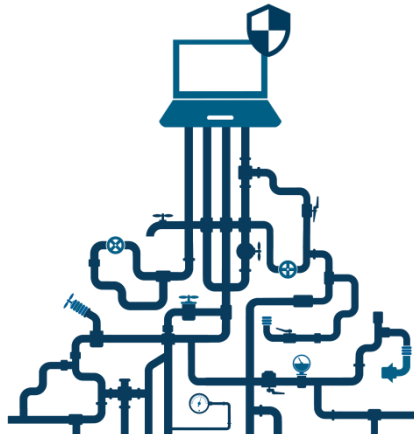
יוני 2019

סיוון התשע"ט

המחלקה להנדסת תוכנה

לוח קונפיגורציות בטכנולוגיות אינטרנט

Web Configurations Dashboard



חיבור זה מהווה חלק מהדרישות לקבלת

תואר ראשון בהנדסה

מאת

אלון שוורץ ואלכס לוינזון

מנחה אקדמי: ד"ר ראובן יגל	אישור: מאשר	תאריך: 30.6.19
אחראי תעשייתי: מר עמיאל אביעזר	אישור: מאשר	תאריך: 30.6.19
רכז הפרויקטים: ד"ר אסף שפנייר	אישור:	תאריך:

תקציר

שיגורים וניסויים של מערכת הביטחון מבוקרים ונשלטים באמצעות מערכות העושות שימוש בקבצי פרמטרים רבים המגדירים את כלל הפעילויות הרלוונטיות המתרחשות במהלך ניסוי, החל ממידע שהועבר מחיישנים ועד העמדות המפעילות או צופות בפעילות בזמן אמת. שליטה ובקרה על נתונים אלו הינה חיונית לצורך קיום הפעילות והפקת המרב. מבנה של קבצים אלה לרוב מסובך וקשה להבנה ועריכה ישירה. מסיבות אלו, נדרשת מערכת שתפשט את הנתונים עבור המשתמש, כך שיוכל לבצע שינויים באופן מבוקר ואמין. בנוסף, יש צורך לנהל את קבצי הפרמטרים בצורה יעילה כך שניתן יהיה לחזור לגרסאות קודמות.

דו"ח זה מתעד את פיתוח המערכת העונה על צרכים אלו.

מערכת ניהול הפרויקט

הערה: המאגר פרטי לפי דרישת האחראי התעשייתי, לקבלת הרשאות גישה למאגר הקוד וליומן הפגישות נא לפנות לאחד מכותבי דו"ח זה.

#	מערכת	מיקום
1	מאגר קוד	https://github.com/jce-il/web-configuration-dashboard
2	יומן	https://github.com/jce-il/web-configuration-dashboard/wiki/Meetings-Summary-&-Project-Planning
3	ניהול פרויקט	https://github.com/jce-il/web-configuration-dashboard/projects/1
4	סרטון גרסה סופית	https://www.youtube.com/watch?v=-yLYzwMRaI0&feature=youtu.be

הצהרה: הפרויקט נעשה בהנחיית ד"ר יגל ראובן במחלקה להנדסת תוכנה, עזריאלי – המכללה האקדמית להנדסה ירושלים.

הדו"ח מציג את עבודתנו האישית ומהווה חלק מהדרישות לקבלת תואר ראשון בהנדסה. אנו מבצעים את הפרויקט כזוג עקב היקפו והיכולת לכיסוי רב יותר בעת ביצוע בדיקות, תכנון ומימוש הרכיבים השונים של המערכת בצורה מקיפה ויעילה - מאשר שהיה עושה זאת מפתח בודד.

לכל אחד מאתנו הייתה תרומה שווה לפרויקט.

תודות

בראש ובראשונה נרצה להודות לד"ר יגל ראובן, המנחה האקדמי, שלאורך כל הדרך עזר ותמך בנו רבות ועל כך שהעניק לנו את ההזדמנות לבצע את פרויקט הגמר עבור חברת רפאל ומערכת הביטחון בידיעה שהמערכת שאנו מפתחים תורמת תרומה משמעותית לביטחון המדינה.

נרצה להודות גם למנחה התעשייתי, מר עמיאל אביעזר, אשר הנחה אותנו מבחינה מעשית לאורך כל תהליך הפיתוח והיה תמיד זמין לשאלות שעלו ונתן טיפים ועצות לגבי תהליך הפיתוח.

לבסוף, תודה גם לד"ר שפנייר אסף, רכז הפרוייקטים, על הזמינות והמענה המקצועי והיעיל על כל שאלה.

תוכן עניינים

3.....	תקציר.....	
5.....	תודות.....	
6.....	תוכן עניינים.....	
7.....	מילון מונחים, סימונים וקיצורים.....	
8.....	מבוא.....	1.
9.....	תיאור הבעיה.....	2.
10.....	תיאור הפתרון.....	3.
10.....	תיאור המערכת.....	3.1
10.....	אפשרויות המערכת.....	3.2
12.....	אופן מימוש הפתרון – צד לקוח.....	3.3
16.....	אופן מימוש הפתרון – צד שרת.....	3.4
18.....	בדיקות.....	4.
20.....	סקר שוק והשוואה לספרות.....	5.
20.....	סקר שוק.....	5.1
21.....	השוואה לספרות.....	5.2
22.....	סיכום/מסקנות.....	6.
23.....	נספחים.....	7.
23.....	רשימת ספרות/ביבליוגרפיה.....	7.1
23.....	תרשימים וטבלאות.....	7.2
26.....	הרחבה על אופן מימוש הפתרון.....	7.3
28.....	תיאור הכלים המשמשים לפתרון.....	7.4
29.....	מסכי המערכת.....	7.5

מילון מונחים, סימונים וקיצורים

1. **XML (eXtensible Markup Language)** - תקן לייצוג נתונים במחשבים. שימוש ב־XML מקל על החלפת נתונים בין מערכות שונות שפועלות על גבי תשתיות שונות. תקן ה־XML לא מגדיר איזה מידע יוצג אלא מגדיר כיצד לייצג מידע באופן כללי. תקן XML שייך למשפחת שפות הסימון.
2. **Framework** (פריימוורק) - הוא מבנה תוכנה המסייע בפיתוח תוכנה ו-'הדבקת' חלקיה השונים לכדי מערכת אחת. הוא מספק פונקציונליות גנרית והפשטה של פעולות, כך שניתן לבצע פעולות מורכבות בדרך נוחה יותר.
3. **NoSQL** - הוא קטגוריה חדשה יחסית של בסיסי נתונים, אשר נותנים פתרון אחסון וגישה למידע שאינו ממודל במבנה טבלאי יחסי אשר נפוץ בבסיסי נתונים יחסיים.
4. **Data Binding** - טכניקה שקושרת בין מקורות מידע של הספק והמשתמש יחד ומסנכרנת ביניהם.
5. **TypeScript** - שפת תכנות חנימית ומבוססת קוד פתוח המפותחת ומתוחזקת על ידי מיקרוסופט. היא מכילה את קבוצת כל פקודות ותחביר JavaScript הפופולרית ומוסיפה עליה טיפוסים סטטיים ותכנות מונחה עצמים מבוסס מחלקות.
6. **Angular Framework** - לפיתוח בסביבת רשת המתוחזקת על ידי גוגל.
7. **JS (JavaScript)** - היא שפת תכנות דינמית מונחית עצמים המותאמת לשילוב באתרי אינטרנט ורצה על ידי דפדפן האינטרנט בצד הלקוח. השפה מרחיבה את יכולות שפת התגיות הבסיסית HTML ומאפשרת בכך ליצור יישומי אינטרנט מתוחכמים יותר.
8. **MongoDB** - בסיס הנתונים המוביל בעולם בקטגוריית NoSQL. בסיס הנתונים נשען על מבנה של מסמך בניגוד לבסיסי נתונים טבלאיים העובדים מעל טבלאות מקושרות. מבנה המסמכים עובד מעל מימוש של JSON.
9. **JSON** - (JavaScript Object Notation) הוא פורמט טקסטואלי, הקריא לאדם, המיועד להעברת מבני מידע המורכבים מזוגות של מפתח-ערך. השימוש העיקרי של הפורמט הוא להעברת מידע בין שרת לצרכן כתחליף לפורמט XML.
10. **Agile** - גישה בהנדסת תוכנה המניחה שפיתוח תוכנה הוא ביסודו בעיה אמפירית, ולא ניתן לפתור אותה בשיטות המתבססות על חיזוי או תכנון. הגישה קובעת שפיתוח תוכנה הוא פיתוח מוצר חדש ומתייחסת אליו כמשחק של שיתוף פעולה מכוון מטרה. הגישה הזריזה לפיתוח תוכנה מניחה שלא ניתן להגדיר במלואה תוכנה מסוימת קודם לפיתוחה בפועל, ומתמקדת במקום זאת בשיפור יכולתו של הצוות לספק תוצרים במהירות ולהגיב לדרישות העולות תוך כדי הפיתוח.
11. **Code review** - הליך בו קוד המקור של התוכנה נבדק לגילוי הפרה של כללי אצבע בתכנות, טיפול בשגיאות והיבטים אחרים הקשורים לאיכות הקוד.
12. **צד שרת** - מונח המתייחס לפעולות המבוצעות על ידי השרת במערכת שרת-לקוח. לרוב, שרת היא תוכנת מחשב, אשר רצה על גבי מכונת שרת מרוחקת, ונגישה מהמחשב או תחנת העבודה המקומית של המשתמש.
13. **צד לקוח** - באינטרנט, המחשב הביתי מהווה "לקוח" שמקבל את המידע שהגיע מהשרת, ומציג אותו למשתמש באמצעות תוכנה ייעודית, לרוב דפדפן. המושג "תכנות בצד הלקוח" מתאר את הפעולות שמבצעים הדפדפן ושאר חלקי המחשב של הלקוח, כדי להציג על צג הלקוח את הדף שהגיע מהשרת.
14. **GUI** - ממשק משתמש גרפי הוא ממשק משתמש לתוכנה או לאתר אינטרנט, המבוסס על עיצוב גרפי (להבדיל מעיצוב טקסטואלי בלבד) של המסך המוצג למשתמש.
15. **Node.js** - מפרש תוכנה לשפת JavaScript שנכתב ב C++. הוא תוכנן לכתיבת אפליקציות רשת סקילאביליות כגון שרתי HTTP.
16. **Express.js** - ספריית קוד פתוח מבוססת MVC שנוצרה עבור Node.js.
17. **"ניסיון"** - מכונה בדו"ח זה גם כ-"פרויקט", הינו אוסף של קצבי פרמטרים המתקבלים כקלט עבור המערכת שפתחנו ובהם משתמש המערכת צופה או עורך.

1. מבוא

חברת רפאל, אשר הוקמה בשנת 1948, הינה חברה ממשלתית ישראלית שעוסקת בפיתוח ובייצור של אמצעי לחימה מתקדמים, לרבות טילים עבור מערכת הביטחון וצה"ל.

בחלק מפיתוח אתר שליטה ובקרה לביצוע ניסויים ושיגורים של מערכת הביטחון בבסיס פלמחים, מפעילה החברה מערכת אשר מאפשרת שליטה ובקרה על אוסף רחב של פרמטרים המגדירים את כלל הפעילויות המתרחשות תוך כדי הפעילות, לרבות חיישנים, מחשבים, תקשורת, אזורים גאוגרפיים, אטמוספירה, מזג אוויר ועוד. במהלך כל ניסוי, מעורבים מערכות ומשאבים רבים אשר מופעלים ע"י מספר רב של בעלי מקצוע ומערכות אחרות בזמן אמת. מערכת זו חיונית ביותר לצורך שליטה על כלל הפרמטרים המוזנים למערכות שלוקחות חלק בפעילות, תוך כדי הפעלתן.

קבצי הפרמטרים מוזנים למערכות השונות בתצורת XML.

כיוון שישנו צורך תמידי בתחזוקה ועריכה של קבצי הפרמטרים לפני כל שיגור ו/או ניסוי שמתבצע, פותחה בחברת רפאל מערכת המאפשרת לבצע פעולות אלו. לרוב, הפעולות הנדרשות הן טעינת קבצים רלוונטיים לניסוי, הזנת או שינוי ערכים, הוספה או מחיקה של שדות, בדיקת תקינות לשינויים שבוצעו ושמירת כלל השינויים.

נדגיש כי לפני או במהלך כל ניסוי או שיגור שמתבצעים, מעורבים קבצי פרמטרים רבים המכילים לעיתים אלפי ערכים. קבצים אלה בנויים בצורה לא ידידותית למשתמש ואינם נוחים לעריכה בצורה ישירה – כלומר לא מאפשרים ביצוע פעולות עריכה דרך הקובץ עצמו, דבר שמעלה את הסיכוי להזנת נתונים שגויים או אף שיבוש ערכים/מפתחות קיימים.

בפרויקט זה פתחנו מערכת חדשה אשר נותנת מענה משופר ומודרני לניהול ותחזוקה של קבצי הפרמטרים ואף מאפשרת למשתמש לבצע פעולות שלא היו קיימות עד כה. המערכת מבוססת על גבי סביבת פיתוח WEB וכוללת צד שרת וצד לקוח, הינה גמישה לשינויים תכנותיים ותומכת בפיתוחים ושדרוגים נוספים שידרשו בעתיד. המערכת מיועדת להמשך פיתוח בחברת רפאל לצורך התאמה וסנכרון עם המערכות הצבאיות שלא ניתן היה לקבל אליהן גישה תוך כדי פיתוח פרויקט זה.

2. תיאור הבעיה

בשירות מערכת הביטחון וחברת רפאל קיימת מערכת מיושנת שפותחה בשפת JAVA ובה ישנן מספר בעיות:

המערכת מותקנת מקומית

- עבור כל מחשב שעליו רוצים לעבוד עם המערכת, יש צורך להתקין אותה מקומית.
- אין שליטה על מספר העתקים זהים או קבצים דומים אך עם תוכן שונה.
- כל קבצי הניסויים מאוגדים בתיקיות רשת ולא במסד נתונים, כלומר קיימת גישה ישירה לקבצים ולא קיימת מערכת לניהולם – כך שהסיכון לאיבוד ו/או פגיעה בקבצים קריטיים הינו עצום.

ממשק המשתמש (GUI)

הממשק מיושן, מסורבל, אינו מודרני וחסרות בו אפשרויות רלוונטיות למשתמש אשר יוכלו להקל עליו בעת שימוש במערכת, כגון:

- הצגת קובץ ה XML בפורמט המקורי
- אפשרויות חיפוש של ניסויים, קטגוריות, קבצים ותוכן של קבצים
- מתן אפשרות למשתמש לבנות את מערך קבצי הניסוי בצורה שהוא יבחר בה, כגון יצירת קטגוריות ותת קטגוריות והכנסה לתוכן קבצי XML רלוונטיים.
- יצירת תצוגות מותאמות אישית על ידי המשתמש עבור קבצי XML בודדים או עבור קטגוריה מסוימת הכוללת שליטה על אופן חלוקת תוכן הקובץ בתצוגתו (האם למשל יוצג בחלוקה לטאבים – tabs לפי תגיות ראשיות או ע"י רשימה של תגיות ראשיות).
- שליטה של המשתמש על אילו שדות בקבצי ה XML ניתן יהיה לצפות.
- התאמת התצוגה ע"י שינוי סוג הפונט, הגודל והצבע של הטקסט, התאמת התצוגה לסוגי מסכים / מקרנים שונים.

איכות הקוד

המערכת הקיימת הנה בשימוש מספר רב של שנים ופותחה במשך הזמן ע"י צוותי פיתוח שונים אשר התחלפו כתוצאה מאילוצים שונים. בשל כך המערכת מכילה קבצים רבים המכילים עשרות אלפי שורות קוד, מה שגורם לקוד מסורבל, מאוד לא גמיש לשינויים ולא מתועד. מצב זה גורם לכך שנדרשים משאבים רבים (מפתחים, זמן) עבור כל שינוי ולו הקטן ביותר שרוצים לבצע. מבחינת חברת רפאל, זוהי הבעיה העיקרית שלשמה דרוש פתרון ופרויקט זה בא לתת עבודה את המענה.

מעבר לכך, הגדרנו בעיה עיקרית בהנדסת תוכנה:

כלליות (גנריות) ופיתוח תשתית קוד גמישה לשינויים

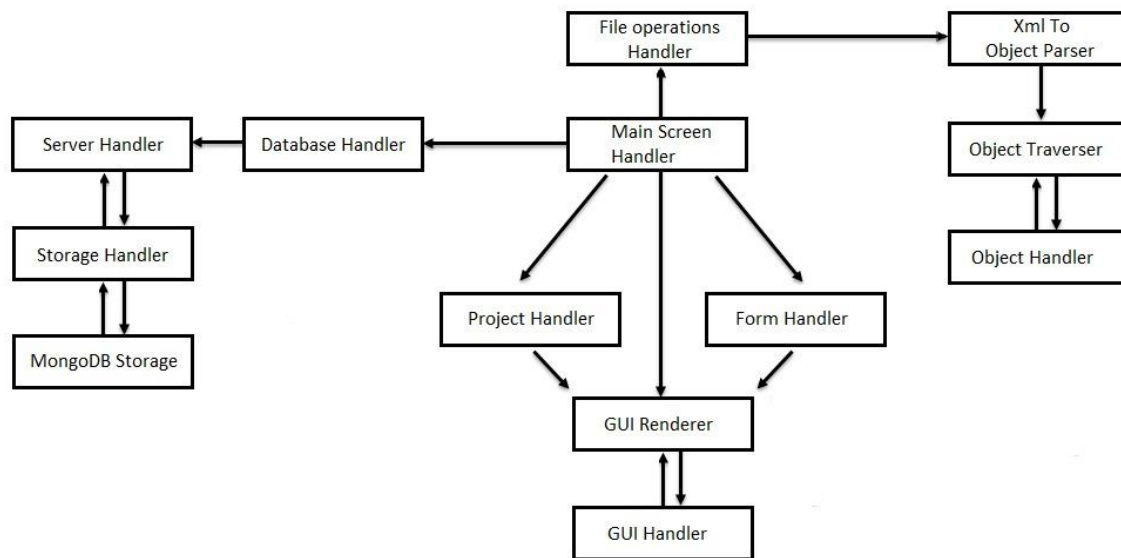
- כיצד לפתח מערכת מותאמת, אשר תתאים לכל סוג של קובץ XML, כאשר מספר הקבצים, גודלם ותוכנם אינו ידוע.
- כיצד לפתח ממשק וחווית משתמש אשר יאפשרו להתאים את המערכת כך שתיתן מענה לשימושים ספציפיים, כאשר השימושים הללו נתונים לשיקול דעתו של המשתמש.
- כיצד מפתחים מערכת מבחינה הנדסית, אשר מצד אחד תהווה בסיס מספק לחוסרים שתוארו לעיל, אך מצד שני תאפשר למפתחים המשך פיתוח קל ויעיל ותהיה גמישה במידה רבה לשינויים ככל שידרשו.

3. תיאור הפתרון

הפתרון שלנו הוא פיתוח מערכת דינמית המבוססת על גישת שרת-לקוח, אשר תאפשר למשתמש ליצור פרויקטים המורכבים מקבצי XML. בכל פרויקט ניתן לחלק את הקבצים לקטגוריות ותתי קטגוריות, ליצור תצוגה מותאמת של הקובץ, לבצע עריכה מלאה של הקבצים ולשמור את הפרויקט למסד נתונים מסודר. בנוסף, המערכת תאפשר לטעון ניסוי קיים ולשחזר כל ניסוי לגרסאות קודמות שלו.

3.1 תיאור המערכת

המערכת פותחה באופן מודולרי ועם תיעוד רב, כך שניתן יהיה להמשיך את הפיתוח בצורה נוחה. חשוב לציין כי על מנת להפוך את המערכת למבצעית, יש צורך בחיבור למספר מערכות צבאיות – מה שדורש עוד המשך פיתוח.



תרשים זה מתאר את המערכת, אשר מורכבת ממספר אלמנטים עיקריים: **Xml To Object Parser** המאפשר עבודה עם הנתונים השמורים בקובץ ה XML על גבי פלטפורמת WEB, **Object traverser** המאפשר ריצה על האובייקט המתקבל לאחר ההמרה, **Object handler** אשר מכיל את רוב הפונקציות שכתבנו המאפשרות לבצע את כלל פעולות העריכה על קבצי ה XML עליהן נפרט בהמשך, **Database Handler** אשר מנהל את הבקשות מצד הלקוח לשרת, **Server Handler** אשר מטפל בבקשות בצד שרת וה-**MongoDB Storage** שמאפשר פעולות אחסון ושליפה של נתונים ממסד הנתונים.

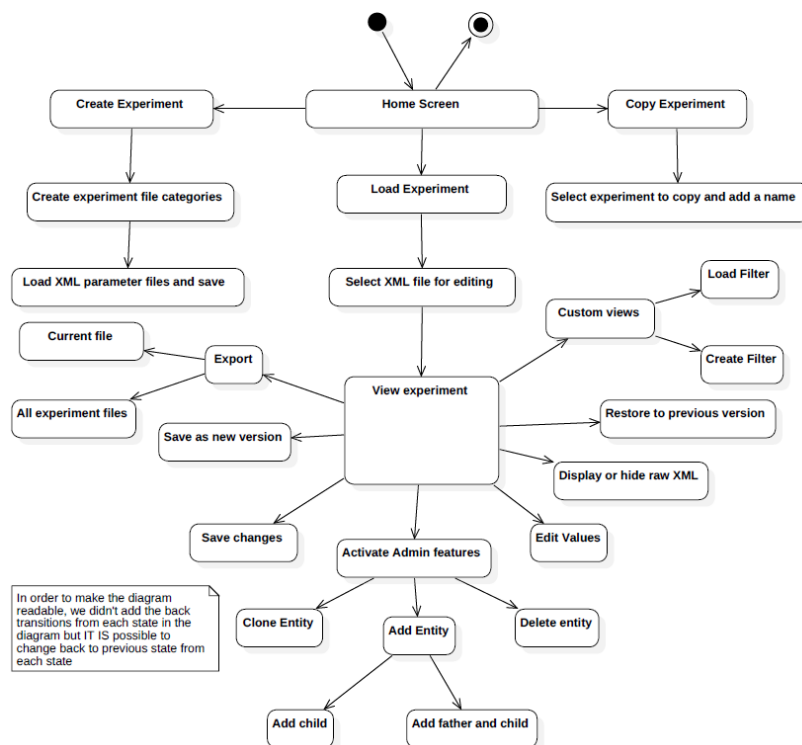
3.2 אפשרויות המערכת

המערכת כוללת את האפשרויות הבאות:

- תמיכה בכל סוגי קבצי ה XML הקיימים בחברת רפאל ללא תלות כזו או אחרת – כלומר גנריות.
- יצירת ממשק המאפשר בניית עץ קבצים וחלוקתם לקטגוריות כך שניתן יהיה לעבוד על מספר רב של קבצים במקביל תחת אותה "קורת גג".
- הצגת תוכן קבצי ה-XML בפורמט טופס אינטרנטי ובמבנה זהה לפי ההיררכיה שהוגדרה בקובץ, כולל חלוקה ל "טאבים" לפי התגיות הראשיות.

- עריכה של קבצי ה-XML הכוללת עדכון ערכים, הוספה, מחיקה ושכפול של שדות.
- צפייה בתוכן קובץ ה XML המקורי תוך הצגה בזמן אמת של שינויים שמתבצעים בקובץ וכן מתן אפשרות לעריכת התוכן דרך תצוגה זו.
- אופציות חיפוש מתקדמות המקובלות בעת שימוש בדפדפן אינטרנטי מודרני.
- יצירת תצוגות מותאמות אישית המאפשרות שליטה על התוכן הנראה למשתמש – כלומר סינון שדות או שינוי שמות לתגיות מסוימות על פי מבנה שהוגדר מראש על ידי יוצר הניסוי.
- יצירת ניסוי חדש כשכפול של ניסוי קיים.
- טעינת ניסוי, כאשר הקובץ אשר יוצג למשתמש לאחר הטעינה הוא הקובץ בו צפה בעת שמירת הניסוי.
- ייצוא קבצי ה-XML לאחר ביצוע השינויים.
- שחזור ניסוי לאחת הגרסאות הקודמות (במידה וקיימות כאלה).
- שמירה וניהול בצד שרת של כלל המידע, הכולל הגדרות מערכת ותצוגה עבור כל ניסוי, תוכן קבצי ה-XML ופרויקטים שהמשתמש יצר על גבי מסד הנתונים (מסוג MongoDB).

המערכת מכילה מספר רב של מצבים, כמתואר בתרשים הבא:



ניתן לראות שרוב העבודה מתבצעת במצב ה **view-experiment** כאשר נטען ניסוי במערכת. **Custom views** מתייחס לאופן תצוגת כלל שדות קובץ ה-XML כאשר המשתמש שולט על איזה שדות יראה ובאיזה אופן. בנוסף, בנספחים קיים תרשים **Use-case** אשר מציג את הפונקציונאליות של המערכת מנקודת מבטו של הלקוח וכמו כן **Class Diagram** ו- **Sequence Diagram** אשר מתארים בצורה יותר מפורטת את תהליכי ורכיבי המערכת.

3.3 אופן מימוש הפתרון – צד לקוח

לעיון במסכי המערכת ניתן לפנות לנספחים בפרק 7.5

הפתרון מבחינה הנדסית

מבנה הנתונים העיקרי שבחרנו להשתמש בו בפרייקט הינו עץ. המחשבה על מבנה זה הייתה טריוויאלית כיוון שמבנה של קבצי XML הוא כמבנה של עץ. לצורך כך השתמשנו בספריית JS קיימת בשם **xml2js** הכוללת פונקציה המרה מקובץ XML לאובייקט JSON. את אובייקט זה ניתן לייצג כעץ. בשלב זה יש לנו אובייקט המייצג את הקובץ – אבל איך נדע לאילו שדות לגשת על מנת לערוך את הקובץ, אם אנחנו לא יכולים לדעת מראש אילו שדות קיימים בו?

לצורך כך אנו צריכים לסרוק את העץ שנוצר, ולכן חיפשנו את הפתרון הטוב ביותר לסריקת עץ שכזה.

התלבטנו בין מימוש עצמי של אלגוריתם הסריקה לבין שימוש בספרייה קיימת. לבסוף החלטנו להשתמש בספרייה בשם **traverse**, אשר מבצעת סריקה רקורסיבית לעומק (DFS) של כל איברי העץ ומאפשרת לבצע עריכה מיידית של השדות.

אחד האתגרים הגדולים בשלב זה היה להבין כיצד להציג ולאפשר שינויים בקבצי XML על ידי המשתמש, כאשר מספר הקבצים והמבנה שלהם אינו ידוע, ובנוסף למצוא פתרון כך שיהיה נכון עבור כל סוג של קובץ XML. כמענה לאתגר זה, יצרנו עץ תצוגה, כמבנה נתונים נוסף לאובייקט הXML, המייצג את הקובץ בצורה גמישה יותר לשינויים. גמישות זו מתאפשרת מפני שבחרנו לממש את עץ התצוגה עם מבנה קבוע, כך שבעצם **המבנה גנרי ולא ידוע למבנה סטטי** שקל לעבוד איתו.

מבנה עץ התצוגה

עץ התצוגה מיוצג באמצעות מערך דו ממדי, כך שעבור כל תגית ראשית ברמה 1 בקובץ הXML הקצנו מערך אשר מורכב מאובייקטים שיצרנו מסוג **xmlNode**. כל אובייקט שכזה מכיל את אפשרויות התצוגה של השדה, מספר הזיהוי, ערך השדה, המפתח ואת הרמה שלו בעץ התצוגה. [לפירוט מורחב הכולל צילומי מסך אנא פנו לנספחים]

לאחר יצירת עץ זה, החלטנו להרחיב את הקוד של ספריית **traverse** ולהוסיף גם מספרי זיהוי לשדות, על מנת שנוכל להבחין בין שדות שונים עם תתי שדות בעלי שם זהה, ולוודא כי אנו מבצעים את השינוי בשדה הנכון.

פעולות עריכה וסנכרון

ישנם כעת שני מבני נתונים שונים, אשר צריכים להיות מסונכרנים ביניהם באופן תמידי. ניתן לבצע הוספה, מחיקה ושינוי של ערכים ואם מבני הנתונים לא יהיו מסונכרנים הדבר יגרום לעריכה לא נכונה של הקובץ.

לאחר מספר ניסיונות רב החלטנו לממש את הסנכרון כך שאחרי כל שינוי בודד שהמשתמש מבצע, אותו השינוי יבוצע מיידית גם על אובייקט הXML ומתבצע ריענון של עץ התצוגה כך שמספרי הזיהוי של כל האלמנטים נותרים זהים בין עץ התצוגה ואובייקט הXML.

אלגוריתם העריכה והסנכרון זהה כמעט לחלוטין בין הפעולות השונות (הוספה, מחיקה או עדכון ערך), ולכן נתמקד בהסבר זה רק על הוספה. מעבר לכך, ניתן למצוא בנספחים צילומי מסך של הפעולות.

כאשר המשתמש בוחר לבצע הוספה, עליו ללחוץ קודם כל על השדה אליו הוא מעוניין להוסיף שדות. בדרך זו אנו משיגים את מיקום התגית אליה אנו אמורים לבצע את ההוספה. לצורך המחשת הדברים, נקרא לה כאן תגית ההוספה.

לאחר שהמשתמש הוסיף שדה בודד או בלוק רציף של תגיות (על ידי שכפול תגית או יצירת תגית אב ותגית בן), אנו מבצעים את ההוספה לעץ התצוגה, כאשר התגיות החדשות מתווספות אל סוף תגית ההוספה.

לאחר מכן אנו מעבירים אל פונקציית העריכה והסנכרון את עץ התצוגה החדש (לאחר שהומר למערך חד ממדי. נפרט את הסיבה בהמשך), את אובייקט XML ואת סוג הפעולה. פונקציית העריכה והסנכרון תפעיל את פונקציית traverse, אשר תסרוק את האובייקט הישן, תמצא את תגית ההוספה ותוסיף לה את התגיות החדשות. לאחר מכן, יתבצע עדכון של מספרי זיהוי באובייקט לכל התגיות שמיקומן השתנה. בסיום פעולה זו, עץ התצוגה ואובייקט XML יהיו אמנם זהים – אך מספרי הזיהוי של חלק מהשדות לא, ולכן אנו מבצעים עדכון של מספרי הזיהוי בעץ התצוגה לפי מה שמופיע באובייקט XML.

פעולות אלו מבוצעות במהירות וביעילות מאחורי הקלעים כך שמבחינת המשתמש לא מורגש שהיה שינוי, פרט לשינוי שביקש לבצע.

תצוגה מותאמת

פילטרים

הוספנו אפשרות לתצוגה מותאמת, המבוססת על סינון מתקדם של השדות, כאשר לכל קובץ בניסוי יש מערך פילטרים נפרד. עבור כל תגית, המורכבת מעוד תגיות, הוספנו ערך "כותרת". אם נקביל זאת לעץ, אז כל צומת בעץ הוא כותרת. כאשר המשתמש מגדיר את הפילטר, עליו לבחור שדות כותרת בלבד. לאחר שהפילטר יופעל, עבור כל שדה כותרת הקיים בפילטר – המשתמש יראה רק את השדות הבנים שלו אשר אינם כותרות, בנוסף לשדה הכותרת שנבחר.

במידה והפילטר פועל והמשתמש הוסיף שדות כותרת לקובץ, השדות יתווספו מיידית לפילטר. במידה והפילטר פועל והמשתמש מחק שדות כותרת מהקובץ, השדות ימחקו מיידית מהפילטר.

בפועל, על מנת לשמור את התגיות הנכונות בפילטר, החלטנו שלא לשמור את מספר הזיהוי. מספר זה ישתנה בסבירות גבוהה לאחר הוספה או עריכה של שדות, ובכדי לעדכן אותו למספר הזיהוי החדש נצטרך לנהל מעקב אחר מספרי הזיהוי ולהחזיק עבור כל שדה את מספר הזיהוי החדש וגם את מספר הזיהוי הישן – דבר אשר הרגיש לנו מיותר ולא יעיל. לחלופין, יכלנו לנהל מבנה נתונים אשר מחזיק עבור כל מספר זיהוי, לאיזה מספר הוא השתנה אחרי ביצוע פעולה – אך גם זה הרגיש לנו מיותר ולא יעיל. הדרך בה בחרנו להשתמש היא על ידי שמירת המיקום המדויק של השדה בעץ התצוגה, כלומר שמירת אינדקס ה-i וה-j.

בדרך זו, אנחנו מסננים את הקובץ על בסיס מיקומי השדות. כאשר מתבצעת הוספה או מחיקה של שדות, אנחנו כבר יודעים את המיקום המדויק של ההוספה או של המחיקה ואת הכמות המדויקת של השדות שהתווספו/נמחקו – וכאשר יש לנו את המידע הזה, עדכון הפילטר הופך לפעולה פשוטה, מהירה ויעילה יותר מהחלופות שהוצגו.

מעבר לכך – במידה והמשתמש ביצע כל הוספה או מחיקה כלשהי לקובץ, אנו מבצעים עדכון של כל הפילטרים של אותו הקובץ, כך שהפילטרים תמיד נשארים מעודכנים, לא משנה איזה שינויים בוצעו בקובץ ולא משנה אם הפילטר פעיל כרגע או לא.

תצוגת מותאמת פר קובץ

כחלק מהמשך פיתוח עתידי של המערכת, נדרשנו להוכיח יכולת ביצועית ליצירת ממשק ייחודי עבור כל קובץ שיוגדר כיוצא דופן הן מבחינת המשתמש והן מבחינת המפתח. להרחבה בנושא אנא פנו לפרק 7.3 בנספחים

תלויות בין קבצים

במקביל להוכחת היתכנות למימוש של התצוגה המותאמת, נדרשנו להראות יכולת ליצירת חיבורים לוגיים בין קבצי XML, שאותם יגדיר המשתמש או המפתח. לדוגמא, קשרים לוגיים אפשריים בין קבצים: (1) בעת שכפול של תגית כלשהי עם ערך ייחודי, התגית החדשה שתיווצר תקבל באופן אוטומטי ערך ייחודי עברה. (2) בעת יצירה או שכפול של תגיות קיימות, ייווצרו תגיות חדשות בקובץ אחר כאשר המידע שאותו הן מגדירות מועתק מהקובץ הראשון לשני.

בכך בעצם נוצר קשר לוגי בין קבצים. **דוגמא מעשית אפשרית לשימוש:** ברשות המשתמש שני קבצים, קובץ ראשון המתאר מחשבים וקובץ שני המתאר מסכים. לכל מחשב ניתן להתאים רק מסך בודד וההיפך. לכן, בעת יצירה של תגיות מחשבים חדשות, המערכת תייצר ערך id ייחודי עבורן ותציב אותו בשדה המתאים, ובנוסף המערכת תייצר באופן אוטומטי בקובץ של המסכים תגית מסך חדשה עם ערך ה id שהוצב בתגית המחשב.

כיוון שבכל רגע נתון יש ברשות המפתח את המידע על כל הקבצים הטעונים במערכת, כולל מיקומים של כל התגיות בכל קובץ, תמיכה ביצירת קשרים לוגיים שתוארו לעיל אפשרית ואף הדגמנו יכולות זו על חלק מהקבצים שיש בהם שימוש בחברת רפאל. אנו מימשנו את הקשרים להדגמה באמצעות מידע המפתח מזין למתודות אך ניתן לקבל מידע זה גם דרך ממשק המשתמש ולהגדיר את ההתאמות לפיו.

נדגיש כי לא נדרש לתחזק מבנה נתונים נוסף מעבר למה שפתחנו עבור המערכת – מה שמותיר את המערכת גמישה יותר לשינויים עתידיים.

חלוקת הקבצים לקטגוריות

יצרנו מסך ייעודי ליצירת ניסוי, ובו על המשתמש ליצור את הקטגוריות לפיהן קבצי XML יחולקו, ולהעלות עבור כל קטגוריה את הקבצים השייכים לה. בשלב יצירת הקטגוריות ניתן גם למחוק קטגוריות או קבצים שנוצרו / נוספו בשוגג על ידי המשתמש. אנו שומרים את הקטגוריות, תוכן והגדרות התצוגה שלהן בתוך מערך. מבנה הנתונים נבנה כעץ – שהוא מבנה נתונים טריוויאלי לייצוג קטגוריות וקבצים.

יצרנו אובייקט בשם Category והעץ מורכב מאובייקט זה, כפי שניתן לראות בתמונה הבאה:

```
▼ (2) [Array(2), Array(1)] ⓘ
  ▼ 0: Array(2)
    ▼ 0: Catagory
      ▶ file_objects: (4) [FileObject, FileObject, FileObject, FileObject]
      ▶ files: (4) [File, File, File, File]
      level: 1
      name: "some category"
      show_files: "-"
      show_selected_category: "-"
      ▶ __proto__: Object
    ▼ 1: Catagory
      ▶ file_objects: [FileObject]
      ▶ files: [File]
      level: 2
      name: "another"
      show_files: "+"
      show_selected_category: "-"
```

אובייקט זה מכיל את שם הקטגוריה, הרמה שלה, הגדרות תצוגה, מערך קבצים, המכיל את הקובץ המקורי שהועלה (בשימוש רק ביצירת ניסוי חדש) ואובייקט שיצרנו בשם fileObject – אשר מכיל מידע רב על הקובץ וייצוגים נוספים של הקובץ. נפרט על אובייקט זה בהמשך.

הדרך בה בחרנו לממש את הקטגוריות אפשרה לנו לבצע סיווג שלהן **כאשר איננו יודעים דבר על הקבצים או על הקטגוריות הרצויות.**

שמירת מצב נוכחי

כאשר המשתמש עובר בין הקבצים השונים בקטגוריות, הקובץ הרלוונטי נטען מאובייקט הקטגוריות. זאת ההתנהגות הרצויה מבחינתנו, אבל התנהגות זו העלתה בפנינו סוגיה חדשה: מה לעשות במידה והמשתמש ביצע שינויים בקובץ, עדיין לא שמר, ועבר לקובץ אחר להמשך עריכה? במצב הנוכחי, כל פעם שיתרחש מעבר בין קבצים – כל השינויים שעשה באותו הקובץ ימחקו.

בכדי לטפל בנושא זה, החלטנו לבצע שמירה באופן מקומי של כל הנתונים הרלוונטיים לקובץ הטעון לפני החלפה לקובץ אחר. כזכור, בעת לחיצה כלשהי על מיקום בעץ קבצים או בעץ התצוגה אנו באופן מיידי מקבל את האינדקסים של המיקום עליו לחצנו (i,j) – מה שסייע לנו באופן מיידי לאתר את מיקום הקובץ אותו ביקשו לטעון.

כאשר המשתמש יבצע שמירה למסד הנתונים, נשלח לשם את הנתונים העדכניים ביותר, ואם יחליט בסופו של דבר שלא לשמור – המידע ימחק בחזרה למסך הבית של התוכנה או ביציאה מהתוכנה.

לצורך שמירת נתונים אלו, ייצרנו את האובייקט `fileObject`, אשר הוזכר קודם. אובייקט זה מורכב משם הקובץ, הכותרת שלו, עץ התצוגה המעודכן, אובייקט ה XML המעודכן, הפילטרים שלו ונתונים נוספים הקשורים בעיקר לתצוגה. בתמונה הבאה ניתן לראות את מבנה האובייקט:

```
▼ file_objects: Array(4)
  ▼ 0: FileObject
    fileClicked: true
    file_name: "COMM_MAPPING"
    file_string: "<?xml version='1.0' encoding='UTF-8'"
    filter_disabled: true
    filters: []
    selected_filter_name: ""
    title: "Comm Mapping"
    xml_object: {Comm_Mapping: {...}}
    xml_object_max_id: 25
    xml_object_tree: (9) [Array(3), Array(3), Array(3),
```

לפני כל מעבר בין קובץ נוכחי לקובץ חדש, שמרנו את ה"מצב" של הקובץ הנוכחי בצורה מקומית, כלומר עדכנו את ה-`fileObject` (שמייצג את קובץ ה XML) הרלוונטי שנשמר בעץ הקבצים לפי המידע שהמשתמש שינה כגון מצב פילטרים, הסתרה או הצגה של שדות, עדכון שדות שהמשתמש ביצע או במילים אחרות – שמרנו **snapshot** של המערכת כשהיא פתוחה על הקובץ הנוכחי.

רק לאחר מכן הקובץ החדש שהמשתמש בחר להציג – נטען למערכת.

3.4 אופן מימוש הפתרון – צד שרת

המערכת תותקן ע"ג מחשב שישמש כשרת, והגישה אליו תתבצע דרך דפדפן אינטרנט סטנדרטי ע"ג רשת מחשבים סגורה בתוך אתר הלקוח. ניתן יהיה להשתמש במערכת גם על השרת עצמו.

לצורך בניית צד השרת, השתמשנו בעיקר ב-Express.js וב-Mongoose. מזכיר כי עבור בניית מסד הנתונים, השתמשנו ב-MongoDB.

השרת תוכנן על בסיס ארכיטקטורת REST, וקיימים בו שני רכיבים עיקריים: Controller ו-Router. בנוסף להם, יצרנו גם מודל של ניסוי בעזרת mongoose. מודל זה מגדיר איך ניסוי שנשמר בשרת אמור להיות בנוי ומוסיף לנו אפשרויות נוחות לעבודה איתו.

בעת קבלת בקשה לצד השרת, הבקשה מועברת אל ה-Router ומשם מועברת לפונקציה המתאימה בקונטרולר לפי סוג הבקשה והפרמטרים שהתקבלו. בסיום ביצוע הפעולה בצד השרת, מוחזרת תשובה לצד הלקוח – בהתאם לבקשה שהתקבלה ומה שבוצע בפועל.

במסד הנתונים יצרנו collection ובו נאחסן documents המורכבים מהמודל שיצרנו. כך נראה מסמך במסד הנתונים שלנו:

```
{
  "_id": ObjectId("5d14a5560475c04b04b8bccd"),
  "categories": Array
    name: "Some Experiment"
    first_ver_id: "5d14a5390475c04b04b8bcc9"
    version: 5
    latest_ver: true
    prev_ver_exp_id: "5d14a5520475c04b04b8bcc"
    date_of_creation: "Thu, 27 Jun 2019 11:15:34 GMT"
  "viewed_tab_when_saved": Object
    file_location: Object
      i: 0
      j: 0
      k: 1
      tab_location: 1
    v: 0
}
```

השרת שלנו תומך במספר פעולות:

הוספת ניסוי - פעולה זו מתחלקת לשני סוגים: הוספת ניסוי חדש לחלוטין, והוספת גרסה חדשה לניסוי קיים. ביצירת גרסה חדשה, בנוסף למספר הגרסה החדשה, אנו שומרים גם את מספר הזיהוי של הגרסה הקודמת וכן של הגרסה הראשונה של הניסוי. כמו כן, יוקצה לגרסה החדשה מספר זיהוי חדש.

עדכון ניסוי - פעולה זו מעדכנת ניסוי קיים עם הקבצים ששונו בלבד, ומשאירה את מספר הגרסה ללא שינוי.

שכפול ניסוי - פעולה זו משכפלת ניסוי קיים, אך מאפסת את כל המשתנים הקשורים לבקרת הגרסאות ומקצה לניסוי המשוכפל מספר זיהוי חדש. אפשרות זאת נוצרה מכיוון שישנם מקרים בהם יש צורך ליצור ניסוי חדש, אך הוא זהה מבחינת המבנה שלו לניסוי קיים. במקרה כזה, אין טעם ליצור ניסוי חדש ואז להגדיר קבצים וקטגוריות ועדיף לבצע שכפול.

מחיקת ניסוי - פעולה זו מבצעת מחיקה של ניסוי קיים ממסד הנתונים.

שליפת נתונים - מתחלק לשני סוגים: שליפת ניסוי כלשהו לפי מספר זיהוי או שליפת רשימה של ניסויים (הגרסאות העדכניות של כל הניסויים או כל הגרסאות של ניסוי מסוים).

במידה ואחד מקבצי הניסוי היה טעון בזמן ביצוע שמירה למסד הנתונים, ברגע שהניסוי יפתח שוב (טעינת ניסוי), המערכת תיפתח בדיוק על אותו הקובץ והמצב שלו בזמן ביצוע השמירה.

מעבר לפעולות אלו, **מימשנו בקרת תצורה** ללא שימוש בספריות עזר לנושא זה, ובסעיף הבא נפרט כיצד מימשנו:

שחזור ניסוי לגרסה קודמת (בקרת תצורה) -

קיימות מספר גישות שונות למימוש בקרת התצורה. חקרנו את הנושא והחלטנו לממש אלגוריתם בהתאם למסקנות. פירוט על המחקר שלנו והשוואה לספרות ניתן למצוא בפרק 5.2

על מנת שנוכל להסביר איך מימשנו את שחזור הגרסאות, עלינו להתחיל עם מה נשמר במסד הנתונים:

ביצירת ניסוי חדש, נשמרים במסד הנתונים מספר נתוני עזר עבור בקרת הגרסאות: גרסת הניסוי, מספר הזיהוי של הגרסה הראשונה של הניסוי, זמן יצירת הניסוי בצד השרת ומשתנה בוליאני המציין כי זוהי הגרסה העדכנית ביותר. בנוסף, קיים משתנה אשר שומר את מספר הזיהוי של הגרסה הקודמת, אך במקרה זה הוא לא מקבל ערך.

כאשר המשתמש יוצר גרסה חדשה לניסוי זה, נוצר document חדש במסד הנתונים המכיל את הגרסה המעודכנת של הניסוי ואת זמן יצירת הגרסה החדשה, יחד עם השינויים המתבקשים: מספר הגרסה גדל ב-1, הניסוי בגרסה זו מקבל מספר זיהוי חדש ואנו שומרים את מספר הזיהוי של הגרסה הקודמת. בנוסף, אנו מעדכנים את המשתנה הבוליאני בגרסה הקודמת של הניסוי, כך שהגרסה הקודמת כבר לא נחשבת לעדכנית ביותר.

כאשר המשתמש ירצה לבצע שחזור של הניסוי לגרסה קודמת, הוא יראה רשימה של הגרסאות הקודמות יחד עם תאריך היצירה שלהן. בזכות זה ששמרנו בכל ניסוי גם את מספר הזיהוי של הגרסה הראשונה, הרכבת רשימה זו מתבצעת באמצעות שאילתה פשוטה יחסית.

לאחר שבחר את הגרסה אליה יתבצע השחזור, אנו מבצעים בצד השרת שלושה דברים: מוחקים את כל גרסאות הניסוי אשר מספר הגרסה שלהן גדול מהגרסה שנבחרה. מעדכנים בגרסה שנבחרה כי היא כרגע הגרסה העדכנית ביותר. מחזירים אל הלקוח את הניסוי בגרסה שנבחרה.

4. בדיקות

ביצענו את רוב הבדיקות בצורה ידנית לאורך כל תהליך הפיתוח תוך כדי תכנון וכתיבה של תסריטי בדיקה אפשריים עבור החלקים שפותחו באותו הזמן. בכך מנענו בדיקות חוזרות ויכולנו לקבל תמונת מצב על תפקוד המערכת באופן תמידי. לקראת סיום הפרויקט, ביצענו בדיקות ידניות מקיפות, לרבות בדיקות קצה שמטרתן בדיקת יציבות המערכת.

#	מה נבדק	תוצאה צפויה	סטטוס	הערות
צד לקוח				
מסך הבית				
1	יצירת פרויקט חדש במערכת הכוללת הזנת סיסמת מנהל, מתן שם לפרויקט, יצירת קטגוריות ותתי קטגוריות והוספת קבצים לקטגוריות	יצירת פרויקט מתאפשרת כאשר הסיסמא שהוזנה נכונה. קטגוריות וקבצים מתווספים כנדרש, ניתן למחוק קטגוריות או קבצים שהוזנו בטעות. לא ניתן ליצור תתי קטגוריות במידה ולא הוזן שם לקטגוריה הקודמת, שמירה ויצירה מתאפשרת רק כאשר הפרטים הוזנו בצורה חוקית. לאחר סיום הפעולה המשתמש מועבר למסך הראשי שבו קיימים כלל הקבצים והקטגוריות שיצר.	✓	זהו השלב הראשון שעל המשתמש לבצע בעת תחילת עבודתו עם המערכת בפעם הראשונה, לכן היה לנו חשוב ששלב זה מתבצע במערכת בצורה מיטבית וכמו כן הוספנו שיטות וולידציה אשר יאמתו את הפרטים שהזין המשתמש ויאפשרו יצירת פרויקט רק במידה והקלט שהוזן חוקי.
2	טעינת ניסוי קיים	בחלונית שנפתחת מופיעים כל הניסויים השמורים במערכת. לחיצה על אחד מהם תעביר את המשתמש לדף הראשי של הפרויקט.	✓	
3	שכפול ניסוי קיים	בחלונית שנפתחת מופיעים כל הפרויקטים שניתן לשכפלם. המשתמש בוחר פרויקט, מזין שם חדש ומאשר. לאחר סיום הפעולה המשתמש מצליח לטעון את הפרויקט ששוכפל	✓	פעולה זו הנה חלק קריטי במערכת שכן הלקוח יבסס על שכפול פרויקטים קיימים לרוב, ולא ירבה בליצור פרויקטים חדשים
מסך הפרויקט				
4	מזעור והרחבה של קטגוריות, לחיצה על קבצי הפרויקט השונים	לחיצה על שמות הקטגוריות מרחיבה וממזערות אותם בהתאם. לחיצה על שמות קבצי הפרויקט מציגה את תכנם	✓	במעבר בין קבצים, במידה ובוצעו שינויים כלשהם בהם עוד טרם שמירתם, הם יישארו באותו המצב בדיוק שבו הושארו – כלומר השינויים נשמרים.
5	בקובץ טעון, הצגת תוכן שונה של תגיות ראשיות בעת לחיצה עליהן	הטאב שנלחץ מסומן, התוכן של הטאב מוצג בהתאמה. במעבר על טאב אחר מוצג תוכן אחר בהתאמה	✓	בדיקה קריטית – מוודאים גישה לכל חלקי הקובץ, גם אם הוא מאוד גדול.
6	עדכון ערכי שדות בקובץ הנבחר לעריכה	השינויים מתבצעים בהצלחה	✓	יש צורך לוודא כי הפעולות בוצעו בהצלחה. במידה והפעולות בוצעו על שדות לא נכונים, הדבר יוביל לשגיאות
7	הודעה על שינויים שלא נשמרו לאחר ביצוע שינויים בקבצי XML. לאחר שמירה (save changes/save as new version) ההודעה תעלה	מוצגת ומוסרת ההודעה בהתאמה	✓	
8	הפעלת הרשאות Admin באמצעות הזנת סיסמא מתאימה	לאחר הזנת הסיסמא הנכונה נפתחו אפשרויות עריכה נוספות – הוספה, מחיקה ושיכפול של תגיות. בנוסף נפתחה אפשרות למחיקת פרויקט	✓	כרגע הסיסמא מאוחסנת כ string. יש לפתח הגנה חזקה יותר לנעילת אופציות עריכה מתקדמות. בפרויקט זה לא נדרשנו לפתח הגנה חזקה מעבר למה שעשינו.
9	ביצוע פעולות הוספה (אב ובן או רק בן), מחיקה ושכפול של תגיות (בתנאי שסיסמת מנהל מוזנת)	הפעולות מבוצעות בהצלחה על כל אחד מהשדות השונים	✓	יש צורך לוודא כי הפעולות בוצעו בהצלחה. במידה והפעולות בוצעו על שדות לא נכונים, הדבר יוביל לשגיאות
10	לחיצה על Show File תציג/תסתיר בהתאמה את תוכן הקובץ בפורמט המקורי	הקובץ מוצג/ בפורמט המקורי לאחר לחיצה על הלחצן. לחיצה נוספת תביא לסגירתו	✓	המשתמש יכול בכל עת לצפות בתוכן הקובץ בפורמט המקורי בזמן ביצוע השינויים בו
11	לחיצה על Export file/Export experiment files תייצא את הקובץ בפורמט המקורי / קבצי הפרויקט	הקובץ/קבצים מיוצאים בהצלחה. במידה ומדובר על חילוף קבצי הפרויקט, מתקבל קובץ zip כאשר בתוכו הקבצים מסורים בתיקיות לפי הקטגוריות שהוגדרו	✓	
12	לחיצה על Home page תעביר את המשתמש לדף הבית כאשר לפני כן תוצג אזהרה במידה ובוצעו שינויים בפרויקט	המשתמש מועבר לדף הבית. מוצגת בהתאם	✓	בחלק זה קריטי היה לוודא שאכן מוצגת הודעת אזהרה במידה ולא נשמר השינויים.
13	לחיצה על Add filter תאפשר למשתמש לבחור שדות רלוונטיים לסינון. בסיום הפעולה המשתמש יידרש לתת שם לפילטר ולשמור.	לאחר סימון השדות שנבחרו לפילטר, מתן שם לפילטר ושמירה, מוצגים רק השדות שנבחרו במסנן. כמו כן הפילטר נוסף בהצלחה לרשימה	✓	אופציה זו היא חלק ניכר בחוויית השימוש במערכת ולכן חובה לוודא שהסינון עובד בצורה נכונה
14	לחיצה Disable Filter תבטל את הסינון	הסינון מבטל	✓	

15	לחיצה על Load filter תציג את הפילטרים הזמינים. בחירה באחד מהם תטען את הפילטר המבוקש	הפילטרים מוצגים, בחירה באחד מהם טוענת אותו	✓	אופציה זו היא חלק ניכר בחוויית השימוש במערכת ולכן חובה לוודא שהסינון עובד בצורה נכונה
16	בפרויקט ספציפי שבו קיימים הקבצים Display_Stations ו Display_computers תטען תצוגה מותאמת עבור קבצים אלה עם פונקציונאליות ייחודית.	התצוגה המותאמת נטענת וכמו כן הפונקציונאליות עובדת כנדרש	✓	תצוגה מותאמת פותחה באופן ראשוני על מנת להראות יכולות עתידיות של המערכת. יש להמשיך פיתוח על מנת לתמוך בשלל תצוגות מותאמות
17	עריכת שינויים בפרויקט: שינוי שם, שינוי שמות קטגוריות וקבצים, הוספת או מחיקה של קטגוריות או קבצים	השינויים מתבצעים בהתאם	--	טרם נתמך – נדרש פיתוח אך היכולת הטכנית קיימת
18	התאמות GUI: שינוי גודל/סוג/צבע של פונט, החלפת תמונת רקע	השינויים מתבצעים בהתאם	--	טרם נתמך – נדרש פיתוח אך היכולת הטכנית קיימת
צד שרת				
18	יצירת פרויקט חדש במסד הנתונים	הפרויקט מתווסף למסד הנתונים ומקבל מספר מזהה ייחודי	✓	
19	טעינת פרויקט קיים ממסד הנתונים	השרת מציג ללקוח את כלל הפרויקטים השמורים אצלו. ברגע שנבחר פרויקט, השרת שולח אותו ללקוח. במידה ובוצעו שינויים באחד מקבצי הפרויקט, הוא יפתח כאשר הקובץ והטאב עליו עבדו פתוחים להמשך עבודה	✓	זהו פיצ'ר UX מאוד חשוב בעינינו שכן הוא עוזר למשתמש להגיע בדיוק לאותה הנקודה מאיפה שהוא הפסיק את עבודתו האחרונה. על כן, חשוב היה לוודא שפיצ'ר זה עובד בצורה המיטבית.
20	שמירת שינויים שבוצעו בפרויקט קיים	הלקוח שולח לשרת את הקבצים המעודכנים והשרת מעדכן את המסד נתונים בהתאם.	✓	לא מעודכן כל הפרויקט אלא רק הקבצים ששוננו.
21	שמירת שינויים כגרסה חדשה	נשלח עותק חדש של אותו הפרויקט לשרת וכן נוצר חיבור בין כלל הגרסאות הקיימות של אותו הפרויקט לצורך שחזור עתידי	✓	זהו פיצ'ר חשוב במערכת – היכולת לשמור גרסאות הנה קריטית במידה ונעשות טעויות מצד המשתמש.
22	מחיקת פרויקט ממסד הנתונים	הפרויקט נמחק בהצלחה	✓	
23	הצגת גרסאות של פרויקט ספציפי	גרסאות קודמות נשלחות ללקוח. המידע המוצג כולל את זמן ותאריך יצירת העותק	✓	
24	שחזור פרויקט לגרסה ספציפית	השרת מוחק את כל הגרסאות שמעל לגרסה הנבחרת ומעדכן את הגרסה שנבחרה להיות הגרסה הנוכחית	✓	
25	יצירת/שכפול פרויקט עם שם / תוכן זהה	השרת מתריעה על כך שקיים ניסוי עם שם / תוכן זהים	--	טרם נתמך – נדרש פיתוח אך היכולת הטכנית קיימת
26	מחיקת כלל הפרויקטים במערכת	במערכת לא קיימים פרויקטים	--	טרם נתמך – נדרש פיתוח אך היכולת הטכנית קיימת
בדיקות טכניות				
27	הפעלת המערכת על גבי דפדפן Chrome	המערכת עולה ועובדת בצורה תקינה	✓	הדפדפן המומלץ כיום לעבודה הוא Chrome
28	הפעלת המערכת על גבי דפדפן Firefox	המערכת עולה ועובדת בצורה תקינה	✓	
29	הפעלת המערכת על גבי דפדפן Edge	המערכת עולה ועובדת בצורה תקינה	X	המערכת אמנם עובדת אך נרשמת איטיות רצינית במעבר בין מסכי המערכת והחלונות שבה. כתוצאה מכך המערכת אינה שמישה בדפדפן זה. נדגיש כי דפדפן ה Edge עובר התאמות על ידי חברת מיקרוסופט כך שבעתיד לא אמורה להיות בעיה עם דפדפן זה
30	הפעלת המערכת על גבי דפדפן Internet Explorer	המערכת עולה ועובדת בצורה תקינה	X	דפדפן זה אינו נתמך, ובכל מקרה יוצא משימוש
31	הפעלת המערכת ללא גישה לרשת האינטרנט החיצונית	המערכת תעבוד ללא שינוי	✓	לפני ההתקנה הראשונית של השרת, נדרש חיבור לרשת האינטרנט – אך ורק במקרה זה
32	התאמה למסכים עם רזולוציות שונות כגון FHD,HD,SD	המערכת תהיה שמישה על כל סוג של מסך מחשב עם רזולוציית SD ומעלה	--	המערכת עובדת בצורה מיטבית על גבי מסך בגודל 22" עם רזולוציית FHD. במידה ומדובר על מסכים אחרים, ניתן לבצע zoom-out בדפדפן עד אשר המערכת תתאים לגודל המסך. נדרשות פעולות פיתוח עתידיות על מנת לתמוך בגדלי מסכים שונים

5. סקר שוק והשוואה לספרות

5.1 סקר שוק

בנוסף לתוכנה הקיימת היום אצל הלקוח, עליה נכתב בהרחבה בסעיפים הקודמים, קיימות עוד מספר תוכנות לעריכה של קבצי XML דרך טופס המציג את הערכים:

1 Editix

בכדי להשתמש בתוכנה יש צורך ברכישת רישיון שאינו זול. התוכנה מאפשרת גם עריכה של סוגי קבצים נוספים, כגון HTML, dtd ו-Xql.

התוכנה כוללת גם עורך XML עם ממשק עריכה הדומה ל־Notepad. בנוסף קיימת אפשרות ליצירת שדות הטופס מראש כאשר מילוי שלו ייצור את קובץ ה־XML עם השדות שמולאו בלבד.

2 XML Notepad

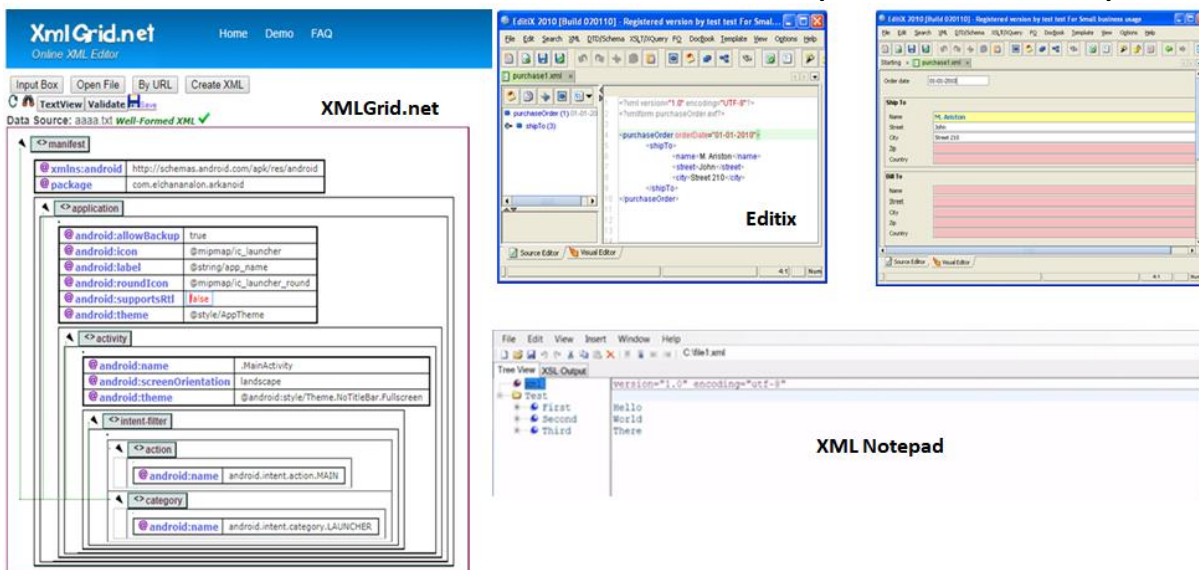
תוכנה חינמית המאפשרת עריכת קבצי XML במבנה של עץ, אך בממשק לא טריוויאלי. התוכנה מותקנת מקומית ואין בה אפשרויות לניהול גרסאות, עבודה מרחוק או ביצוע שינויים בקוד.

3 XMLGrid.net

אתר אינטרנט המאפשר המרה חינמית של קבצי XML לתצוגת טבלאות היררכיות. יש אפשרות לראות את השינויים שלא נשמרו עדיין ועריכת השדות מתבצעת בצורה קלה ונוחה עבור קבצי XML קצרים. מאפשר לבצע בדיקה אם הקובץ הוא בפורמט XML תקני לאחר העריכה. על מנת לערוך קובץ XML ניתן לטעון קובץ או להעתיק את תוכנו לאתר. במידה ונטען קובץ, שמירה של השינויים תוביל להורדת עותק חדש למכשיר. עריכת קבצי XML ארוכים עשויה להיות מסורבלת ולא נוחה, אין אפשרות לבצע שינויים בקוד והאתר עצמו לא מאובטח. אין אפשרות לבצע חיפוש או סינון.

מעבר לתוכנות שהוצגו לעיל, קיימות מספר תוכנות נוספות אך הן בעלות מאפיינים זהים או דומים לתוכנות שהצגנו ואינן תורמות להשוואה. בשל סיבה זו מצאנו לנכון לא לציין אותן ברשימה הנ"ל. כמו כן, נראה כי אף אחת מהתוכנות הקיימות בשוק תוכל להתאים באופן מלא לדרישות הלקוח.

צילומי מסך של שלוש התוכנות שסקרנו



5.2 השוואה לספרות

לפני מימוש מערכת בקרת הגרסאות חקרנו נושא זה, על מנת להבין מהי הדרך הטובה ביותר לממש את הנושא כך שהמערכת תתאים בצורה המיטבית לצרכי חברת רפאל ותשמור על יעילות.

לצורך כך קראנו מספר מאמרים ומדריכים (ניתן למצוא בסעיף הספרות) והבנו שעומדות בפנינו שלוש אפשרויות עיקריות:

(1) עבור כל ניסוי, להחזיק מערך עם כל הגרסאות הקודמות של הניסוי.

יתרונות מרכזיים:

- כל הגרסאות מסודרות באותו הקובץ, לכן ניתן למשוך אותן במהירות.
- חיפוש גרסה מסוימת אמור להיות מהיר יותר משיטות אחרות (חיפוש במערך ממזין)

חסרונות מרכזיים:

- זמן הטעינה גדל עם הזמן (נוצרות גרסאות חדשות וטוענים אובייקט גדול יותר)
- אם אין שימוש רב בגרסאות הישנות, מתבצעת טעינה מיותרת של כל הגרסאות בכל פעם

(2) עבור כל ניסוי, לשמור בכל פעם רק את השינויים שהתבצעו מהגרסה הראשונה/הקודמת, בנוסף למספר הגרסה החדשה.

יתרונות מרכזיים:

- חוסך במקום – לא שומרים את כל הניסוי אלא רק את השינויים
- קל להבין מה ההבדל בין הגרסאות

חסרונות מרכזיים:

- שליפת גרסה מסוימת מהשרת עלולה לקחת זמן – צריך להחיל את השינויים על הגרסה השמורה.
- לא מספיק לנו לדעת שהיה שינוי בקובץ – אנו מוכרחים לתעד את המיקום המדויק של השינוי ותוכנו
- במידה ושומרים רק את הגרסה הראשונה במלואה, בכל פעם צריך לבצע השוואה מול הגרסה הזאת – למרות שאת השינויים ביצענו על גרסה חדשה יותר.

(3) לשמור כל גרסה חדשה של הניסוי בנפרד

יתרונות מרכזיים:

- שליפת ניסוי מתבצעת במהירות – שולפים מהשרת רק את הניסוי המבוקש (ללא גרסאות קודמות ושינויים שצריך להחיל)
- שמירת גרסה מתבצעת במהירות – אין צורך לבצע משהו מעבר לשמירה (כמו למשל הכנסת גרסה למערך הגרסאות הקודמות לפני יצירת גרסה חדשה)

חסרונות מרכזיים:

- תופס מקום רב לאורך זמן (אך זה פתיר)
- במידה ויש שימוש רב בגרסאות הישנות, נצטרך לבצע קריאות מרובות לשרת

לאחר שביררנו מול חברת רפאל מה חשוב להם בנושא זה, הגענו למסקנה ששיטה מס' 3 היא העדיפה מבחינתנו ולכן מימשנו לפי עקרונות שיטה זו.

6. סיכום/מסקנות

כזכור, ברשות חברת רפאל ומערכת הביטחון קיימת מערכת מבצעית אשר מאפשרת שליטה ובקרה על ניסויים המורכבים מקבצי פרמטרים רבים. במהלך פיתוח הפרויקט עמדו לנגד עינינו שלוש מטרות עקריות:

- עמידה בדרישות הלקוח וסיפוק הפונקציונאליות שהגדיר למערכת החדשה, לרבות תמיכה בכלים שהיו קיימים במערכת הנוכחית.
- הוספת חידושים ופיתוחים שאינם קיימים כעת והינם דרישת הלקוח.
- פיתוח מערכת גמישה לשינויים, כאשר סך ההשקעה בעת פיתוח כלים חדשים יהיה קטן יותר משמעותית בהשוואה למאמץ הנדרש לפיתוחים אלו כעת.

בהתאם למטרות אלה, הצלחנו לפתח מערכת עם ממשק משתמש מודרני, העונה על דרישות הלקוח, תוך כדי שימוש בכלי פיתוח עדכניים הקיימים בשוק. בהצלחת המנחה האקדמי שלנו, ד"ר יגל ראובן, בחירת בארכיטקטורת פיתוח של שרת-לקוח בסביבת Web, התבררה כנכונה וסייעה לנו להגיע לתוצאות הרצויות עם נוחות פיתוח מקסימלית.

אחד מהאתגרים שליוו אותנו לאורך תהליך הפיתוח היה לנסות ולשמור על **עיקרון הגנריות**, שלפיו פיתחנו כלים למערכת אשר לא יגבילו הן את המשתמש והן את המפתח לשימוש ספציפי – אלה יתאימו לשימושים שונים ואף יהיו נוחים לשינויים. אתגר זה דרש מאתנו הבנה עמוקה של תהליכים ושלבם בפיתוח תוכנה, תוך שימוש בכלים רבים שרכשנו במהלך תקופת לימודינו לתואר ראשון בהנדסת תוכנה במכללה. על כן, הפרויקט תרם לנו ידע רב והכנה מיטבית עם יציאתנו לשוק העבודה.

מספר מסקנות עלו במהלך העבודה על הפרויקט, וביניהן:

- מומלץ שלא להינעל על דרך פתרון מסוימת למשך תקופה ארוכה במידה והפתרון לא עולה תוך פרק זמן סביר
- יש להשקיע מחשבה מעמיקה על ההשלכות על הרכיבים הקיימים ועל אופן עבודת המערכת אחרי הוספת רכיבים חדשים או ביצוע שינויים משמעותיים ברכיבים הקיימים.

לסיום, נציין שפרויקט זה שונה במקצת מכל פרויקט אחר ששמענו עליו מסטודנטים הלומדים עמנו. המערכת שפתחנו מיועדת להחליף מערכת מבצעית בשירות חברת רפאל ומשרד הביטחון, ועל כן הידיעה שאנו תורמים בדרך זו או אחרת לביטחון המדינה וגם העובדה שהמערכת מיועדת להמשך פיתוח והתממשקות עם מערכות אחרות בחברה, הגבירה את המוטיבציה שלנו להצלחה בכל משימות הפרויקט.

אנו מודים על ההזדמנות שניתנה לנו לעבוד על הפרויקט, שכן בזכותו למדנו רבות, התנסינו, נתקלנו באתגרים, פתרנו אותם, ובעיקר נהנינו מהדרך (:

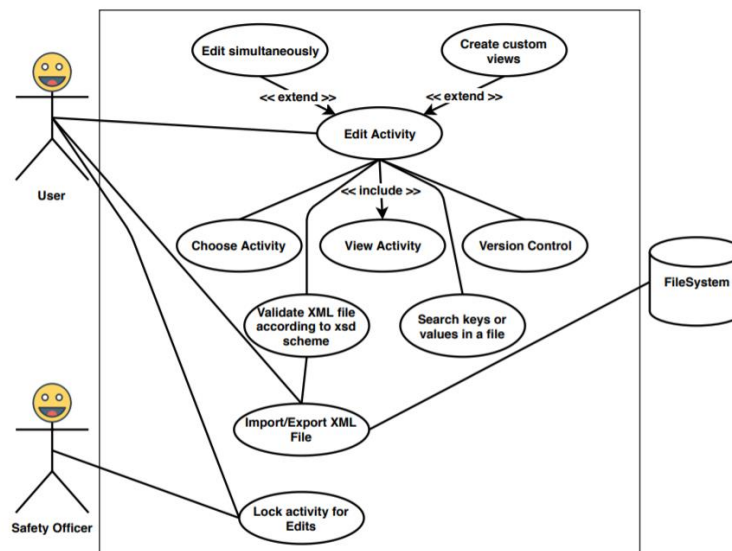
7. נספחים

7.1 רשימת ספרות/ביבליוגרפיה

1. **Angular** - <https://angular.io>
2. **MongoDB** - <https://www.mongodb.com>
3. **TypeScript** - <https://www.typescriptlang.org/docs/home.html>
4. **FireBase** - <https://firebase.google.com/docs/firestore>
5. **Redis** - <https://redis.io/topics/faq>
6. **Amazon DynamoDB** - <https://aws.amazon.com/dynamodb>
7. **Panoply** - <https://blog.panoply.io> – Website that contains comparison reviews about different database platforms.
8. **Db-engines** - <https://db-engines.com> – Same as (7) but the links connects to a specific comparison that we did.
9. **Visual Studio Code** - <https://code.visualstudio.com/> - The IDE that we used for this project development.
10. **REST API Design Book** – Mark Masse, OREILLY publication
11. **Algorithm Design**, J. Kleinberg and E. Tardos.
12. **Gamma, E. et al.** - “Design Patterns – Elements of Reusable Object-Oriented Software” – AddisonWesley – 1995 – reprinted 2004.
13. **Software Configuration Management Handbook** – Alexis Leon, 2005
14. **Configuration Management** – Daniel Galin, 2018

7.2 תרשימים וטבלאות

Use Case Diagram

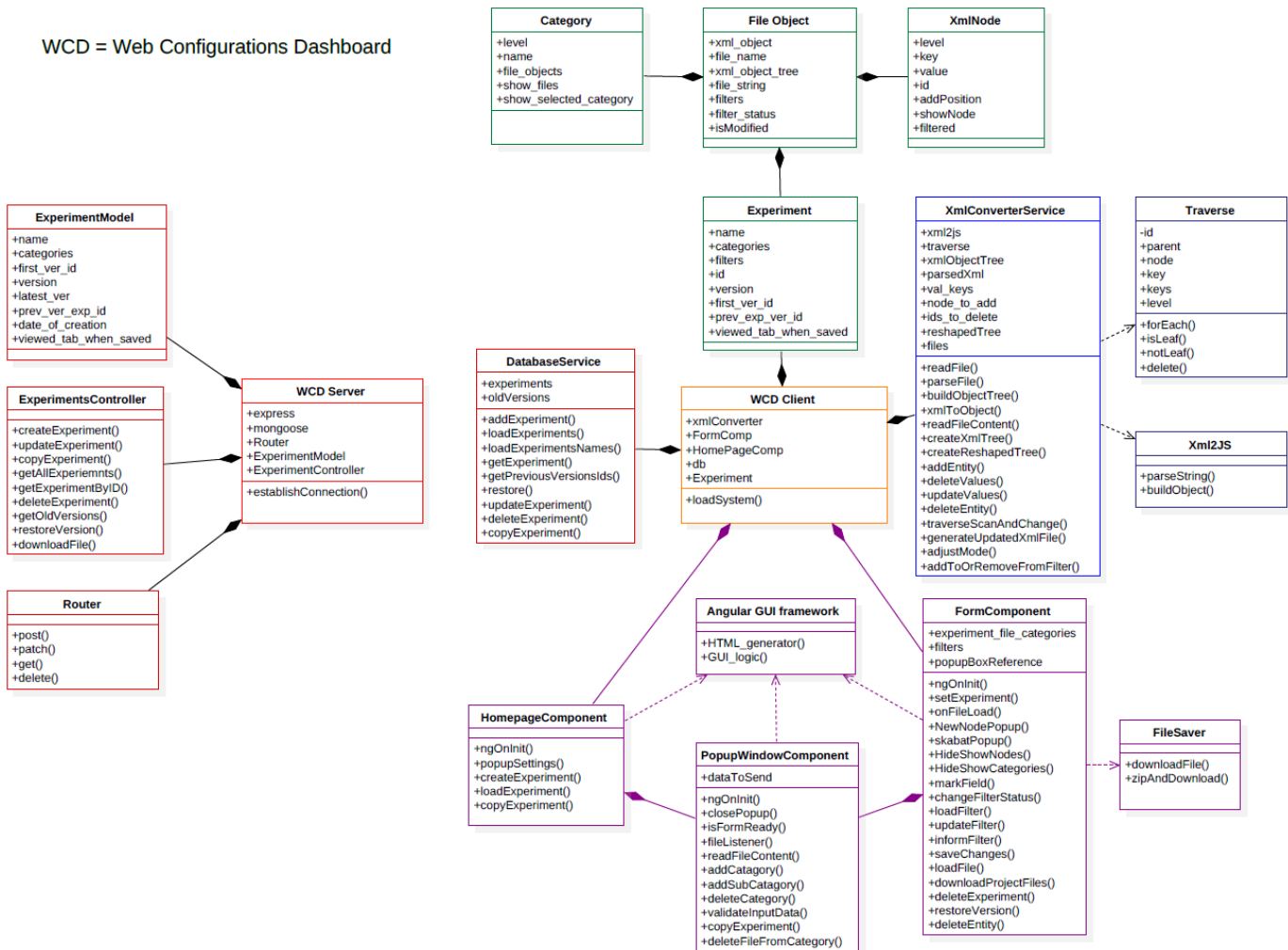


למערכת שני שחקנים עיקריים, המשתמש ומסד הנתונים. רוב הפעולות המתוארות בתרשים דורשות פעולה הדדית ומקבילה הן של המשתמש והן של מסד הנתונים, כלומר המערכת בנויה על עיבוד וצפיה בנתונים. ניתן לראות שרוב הפעולות המפורטות קשורות למניפולציות על מידע, הן מצד המשתמש והן מצד השרת. שחקן נוסף ומינורי הינו קצין הביטחון - שגם הוא מבצע מניפולציות על המידע בכך שהוא מונע עריכה של הנתונים. התרשים מתאר באופן דיי ברור את הפעולות שיהיה ניתן לבצע ואילו שחקנים נדרשים בכל פעולה.

Class Diagram

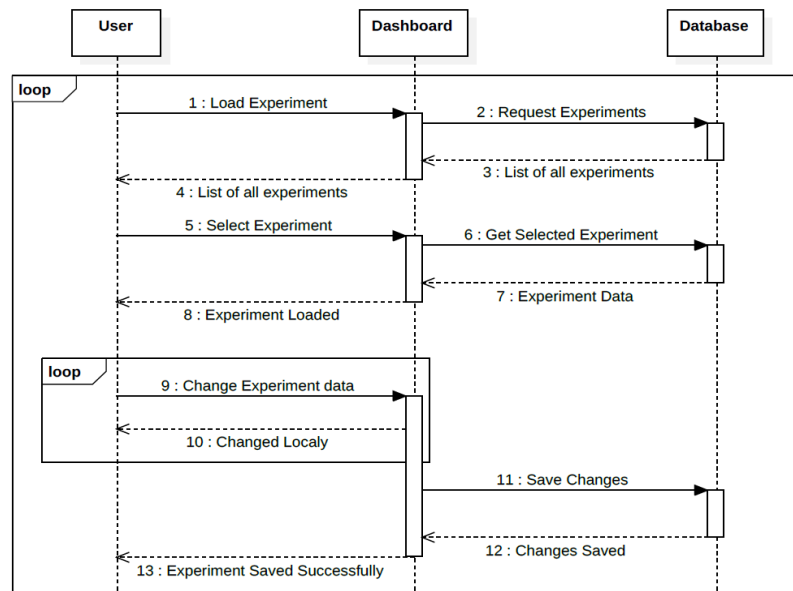
המערכת מורכבת ממספר מחלקות כאשר מחלקות שמכילות את המילה Component מכילות גם את רכיבי ה GUI של אותו רכיב. כלומר, לדוגמה ה- FormComponent מכיל בתוכו הן את השיטות המגדירות את הפונקציונאליות של הטופס והן את ההגדרות הגרפיות שלו.

WCD = Web Configurations Dashboard



Sequence Diagram

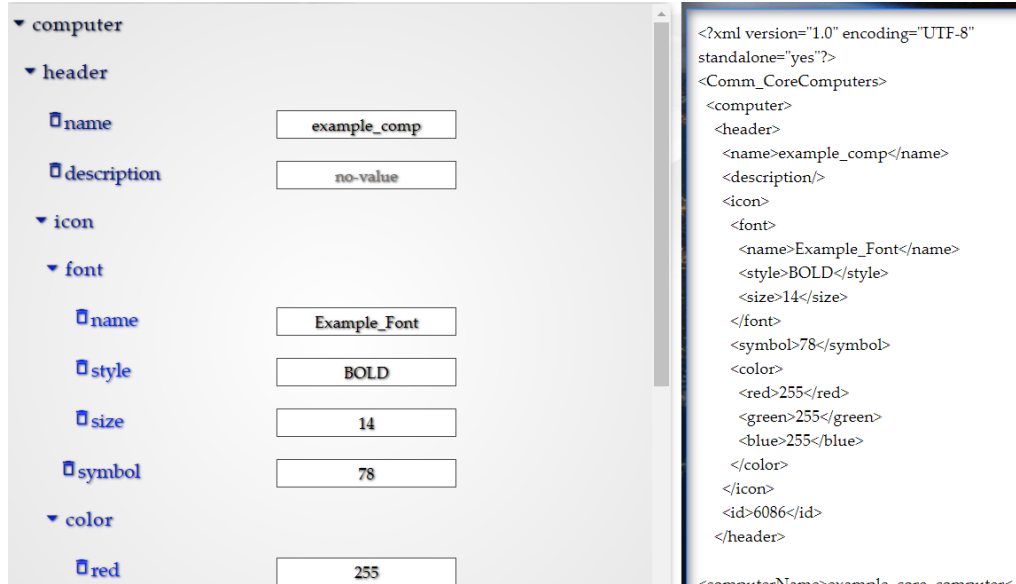
תרשים זה מתאר את אחד השימושים במערכת – טעינת ניסוי, ביצוע עריכה על גבי קבצים הקשורים לניסוי זה ולאחר מכן שמירת השינויים בחזרה למסד הנתונים. ניתן לראות שיש 3 ישויות המעורבות בתהליך: המשתמש, צד לקוח (Dashboard) וצד שרת.



7.3 הרחבה על אופן מימוש הפתרון

עץ התצוגה

ניקח לדוגמה את הקטע הבא מתוך קובץ XML:



The screenshot displays a graphical user interface for editing XML data. On the left, a tree view shows a hierarchy: 'computer' (expanded) contains 'header', 'icon', 'font', and 'color'. Each sub-node has input fields for its properties. For example, 'header' has 'name' (example_comp) and 'description' (no-value). 'font' has 'name' (Example_Font), 'style' (BOLD), 'size' (14), and 'symbol' (78). 'color' has 'red' (255). On the right, an XML editor shows the generated XML code, which matches the structure and values in the tree view.

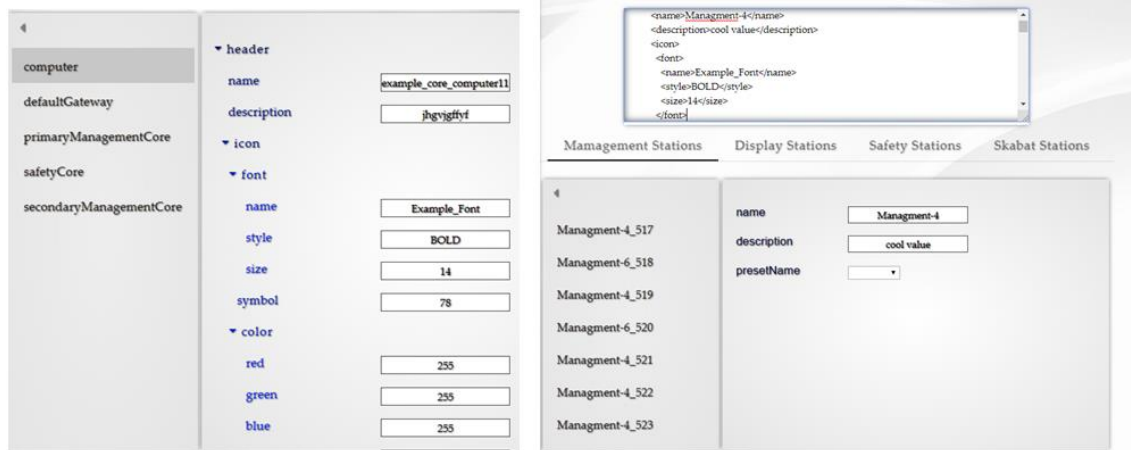
וחלק מעץ התצוגה שנוצר עבור קובץ זה:

```
(5) [Array(23), Array(17), Array(1), Array(1), Array(1)]
▼ ⓘ
  ▼ 0: Array(23)
    ▶ 0: {show_headers: "-", show_nodes: "-", collapse_exp...
    ▶ 1: {show_headers: "-", show_nodes: "-", collapse_exp...
    ▶ 2: {show_headers: "-", show_nodes: "-", collapse_exp...
    ▶ 3: {show_headers: "-", show_nodes: "-", collapse_exp...
    ▶ 4: {show_headers: "-", show_nodes: "-", collapse_exp...
    ▶ 5: {show_headers: "-", show_nodes: "-", collapse_exp...
    ▶ 6: {show_headers: "-", show_nodes: "-", collapse_exp...
    ▼ 7:
      collapse_expand: "Collapse"
      filtered: false
      id: 8
      key: "style"
      level: 5
      show_headers: "-"
      show_nodes: "-"
      ▶ value: ["BOLD"]
      ▶ __proto__: Object
    ▶ 8: {show_headers: "-", show_nodes: "-", collapse_exp...
    ▶ 9: {show_headers: "-", show_nodes: "-", collapse_exp...
```

ניתן לראות כי הרמה של השדה **style** באובייקט היא 5, וזו אכן הרמה שלו בקובץ. בנוסף ניתן לראות כי העץ מורכב ממספר מערכים. מבנה זה מאפשר לנו שליטה טובה יותר על האובייקט ועריכה נוחה שלו, ובנוסף מכיל את אפשרויות התצוגה של כל שדה בקובץ.

תצוגה מותאמת עבור קבצים ספציפיים

לצורך ההדגמה, לקחנו את אחד מהקבצים המיוחדים הללו ויצרנו עבורו תצוגה מותאמת:



מצד שמאל ניתן לראות את התצוגה הכללית לקובץ כלשהו. מצד ימין זוהי התצוגה המותאמת פר קובץ. ניתן לראות שישנה חלוקה לטאבים ראשיים ומשניים, וכמו כן שם הטאבים בעמודה השמאלית מורכב משילוב ערך התגית `name` יחד עם `id`. שליטה על אופן ההצגה של הקובץ וקביעת הפרמטרים והתאמתה לקובץ שיבחר על ידי המשתמש ניתן ליישום על ידי המשך פיתוח של התשתית שיצרנו במערכת. המתודות האחריות על מאפייני ההצגה מסוגלות לקבל כל פרמטר שיקבע המתכנת, או לחילופין, שיתקבל על ידי המשתמש ולפי פרמטרים אלה לקבוע את אופן ההצגה של הקובץ.

אובייקט ה `category` בתוך עץ הקבצים

```
▼ (2) [Array(2), Array(1)] ⓘ
▼ 0: Array(2)
  ▼ 0: Catagory
    ▶ file_objects: (4) [FileObject, FileObject, FileObject, FileObject]
    ▶ files: (4) [File, File, File, File]
    level: 1
    name: "some category"
    show_files: "-"
    show_selected_category: "-"
    ▶ __proto__: Object
  ▼ 1: Catagory
    ▶ file_objects: [FileObject]
    ▶ files: [File]
    level: 2
    name: "another"
    show_files: "+"
    show_selected_category: "-"
    ▶ __proto__: Object
    length: 2
    ▶ __proto__: Array(0)
  ▼ 1: [Catagory]
    length: 2
    ▶ __proto__: Array(0)
```

אובייקט זה מכיל את שם הקטגוריה, הרמה שלה, הגדרות תצוגה, מערך קבצים, המכיל את הקובץ המקורי שהועלה (בשימוש רק ביצירת ניסוי חדש) ואובייקט שיצרנו בשם `fileObject` – אשר מכיל מידע רב על הקובץ וייצוגים נוספים של הקובץ.

7.4 תיאור הכלים המשמשים לפתרון

Angular2+ - דרישה מרכזית בפרויקט הייתה לפתח את המערכת בסביבה מבוססת web, שכן הלקוח ביקש ליצור מערכת web-client אשר תאפשר גישה נוחה לשרת שעליו תותקן המערכת באמצעות דפדפן סטנדרטי בלבד. סביבת פיתוח זו מכילה גמישות תכנותית רבה יותר מבעבר (בהשוואה ל JavaScript) הכוללת בין היתר אפשרות לבצע Data Binding, יצירת ממשקי UI נוחים ומודרניים ויכולות תכנות מתקדמות אשר מבוססות על פונקציות קיימות של שפת Java Script ויכולת מובנת לביצוע בדיקות כגון בדיקות יחידה.

כמו כן גישת הפיתוח בסביבה זו היא **תכנות מונחה עצמים** אשר כוללת יצירת מחלקות, שירותים, ורכיבים וכמו כן שימוש באובייקטים ויכולות כימוס של פונקציות ומשתנים. מהיכרות קודמת עם סביבה זו וחווית תכנות טובה שהייתה לנו איתה, בחרנו לפתח את המערכת איתה מאשר ללמוד Framework חדש.

xml2js – ספרייה של JavaScript המאפשרת המרה של קובץ XML לאובייקט JSON וההיפך. בעצם זוהי ספריית parsing.

Traverse – ספרייה של JavaScript הכוללת פונקציה רקורסיבית המאפשרת מעבר על כלל האלמנטים של אובייקט JSON. כלומר, עבור הפיתוח שלנו הספרייה מאפשרת גישה לכל אלמנט של קובץ XML כלשהוא למחרוזת JSON. בהתבסס על פונקציה זו, פיתחנו את כלל האלגוריתמים שמאפשרים ביצוע פעולות ע"ג קובץ ה XML והן תצוגתו למשתמש.

Bootstrap – על מנת לפתח ממשק מודרני, יעיל וכזה ש"נראה טוב" בעיניי המשתמש, בחרנו להשתמש בספריית ה Bootstrap המקובלת בעולם של עיצוב אפליקציות web ודפי אינטרנט מודרניים. בחירתנו להשתמש בספרייה זו הקלה מאוד על עבודתנו במהלך שלבי העיצוב ובניית חלונות המערכת ולפי דעתנו הצלחנו להגיע לחווית משתמש טובה ולממשק נח. בנוסף, כיוון שהפופולריות שלה רבה, העדפנו להשתמש בה על פני ספריות מתחרות אחרות.

MongoDB - המצב הרצוי מצד הלקוח שימוש במסד נתונים מסוג **NoSQL**, ולכן העדפנו להשתמש במסד נתונים מהסוג הזה.

מבין אפשרויות הקיימות בשוק, חיפשנו סביבה אשר תתמוך במספר דרישות: יכולת עבודה בעמדת מחשב ללא חיבור לרשת האינטרנט – כלומר מסד הנתונים נמצא על המחשב המשמש כשרת, בעלת תיעוד מקיף ככל הניתן, חינומית, מאפשרת התאמה אישית ברמה גבוהה, בעלת API נוח וניתן ללמוד אותה לעומק בזמן קצר יחסית.

לאחר שבדקנו מספר אפשרויות, כדוגמת Amazon DynamoDB, Redis, Firebase ו-MongoDB, הגענו למסקנה כי MongoDB תענה בצורה הטובה ביותר לדרישות הללו ובנוסף לכך, הלקוח המליץ על מסד נתונים זה.

FileSaver - ספריית קוד פתוח אשר באמצעותה אפשרנו הורדת קבצים בצד הלקוח.

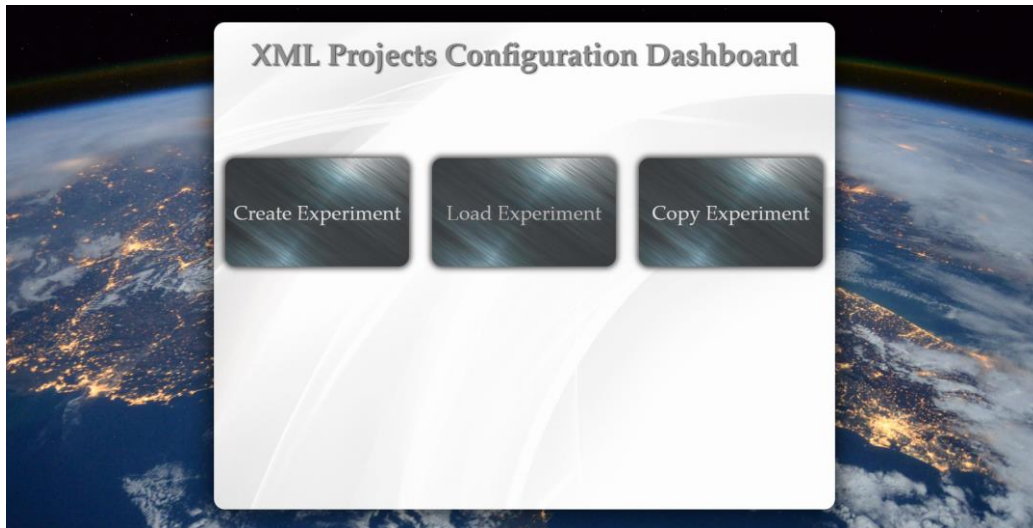
JSZip - ספריית קוד פתוח אשר נעזרנו בה בכדי ליצור תיקיות ולכוון לתוכן קבצים, לצורך הורדת קבצים של ניסוי שלם.

Visual Studio Code – סביבת העבודה בה השתמשנו

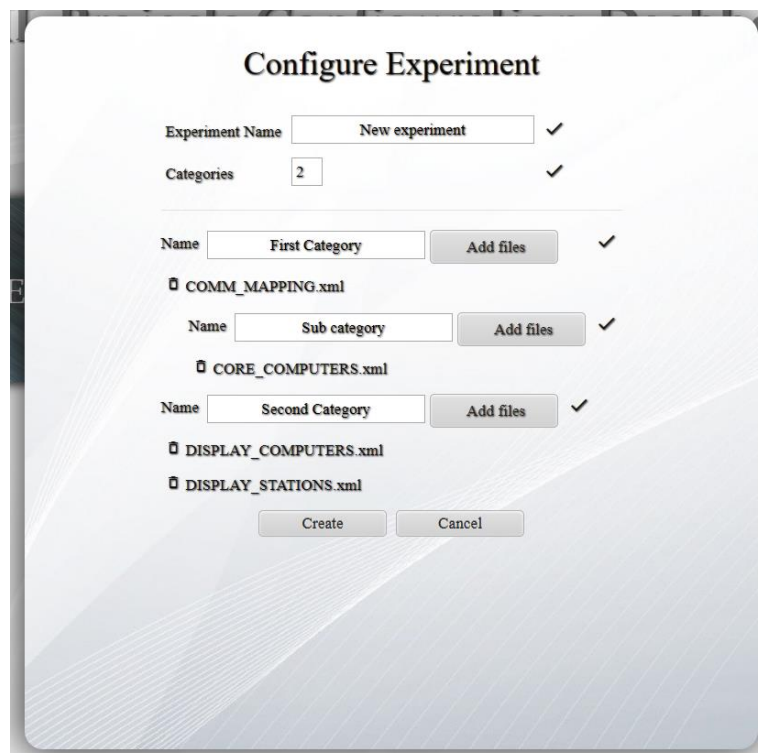
תחזוקתיות הקוד - קוד המערכת פותח בצורה כזאת שמאפשרת גמישות רבה לשינויים מבחינת תוכניתית(קוד) וידידותית למתכנת (חלוקה לתתי מחלקות/ספריות, קבצי תיעוד מלאים של הקוד) כך שמרגע סיום חלקנו בפרויקט זה, יוכלו להמשיך שאר חברי צוות הפיתוח בחברת רפאל להמשיך ולבצע שינויים או שדרוגים בצורה קלה ונוחה במערכת.

7.5 מסכי המערכת

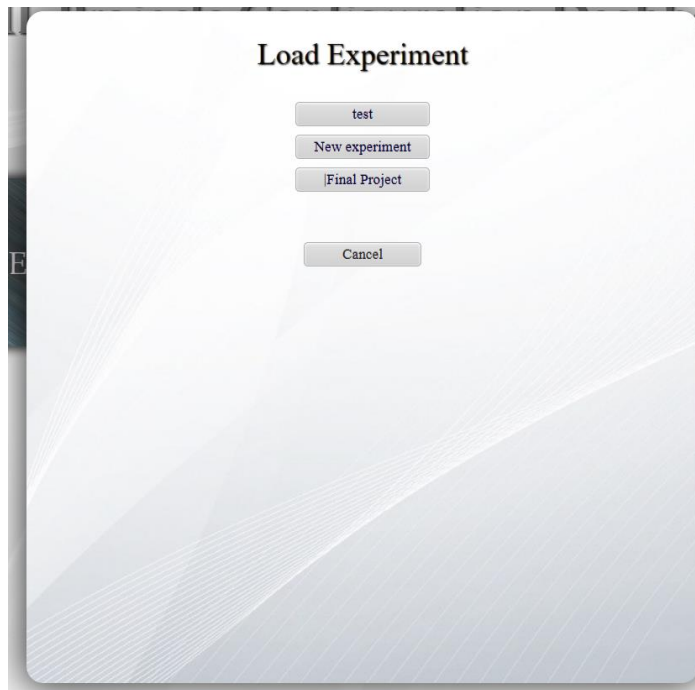
מסך הבית של המערכת. ניתן ליצור פרוייקט חדש, לטעון פרוייקט קיים או לשבפל פרוייקט קיים.



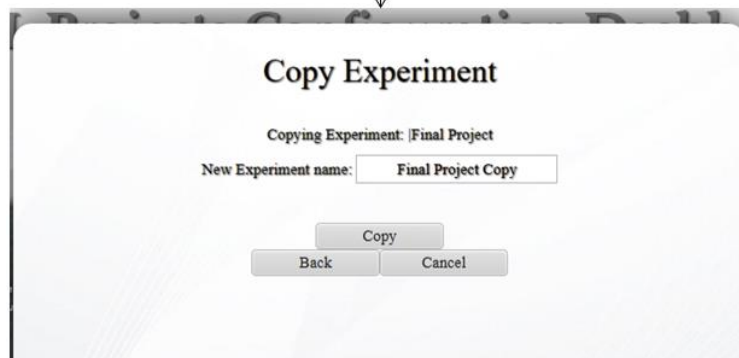
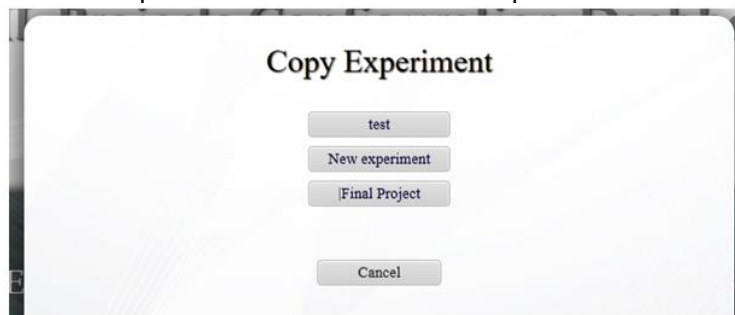
חלון יצירת פרויקט חדש. כפי שניתן לראות, ניתן שם, נוצרו קטגוריות והוספו אליהן קבצים עפ"י בחירתו של המשתמש. לאחר לחיצה על create הפרויקט ישמר יחד עם הקבצים שקושרו אליו במסד נתונים והמשתמש יועבר למסך הראשי של המערכת ובו יוצג הפרויקט שיוצר.



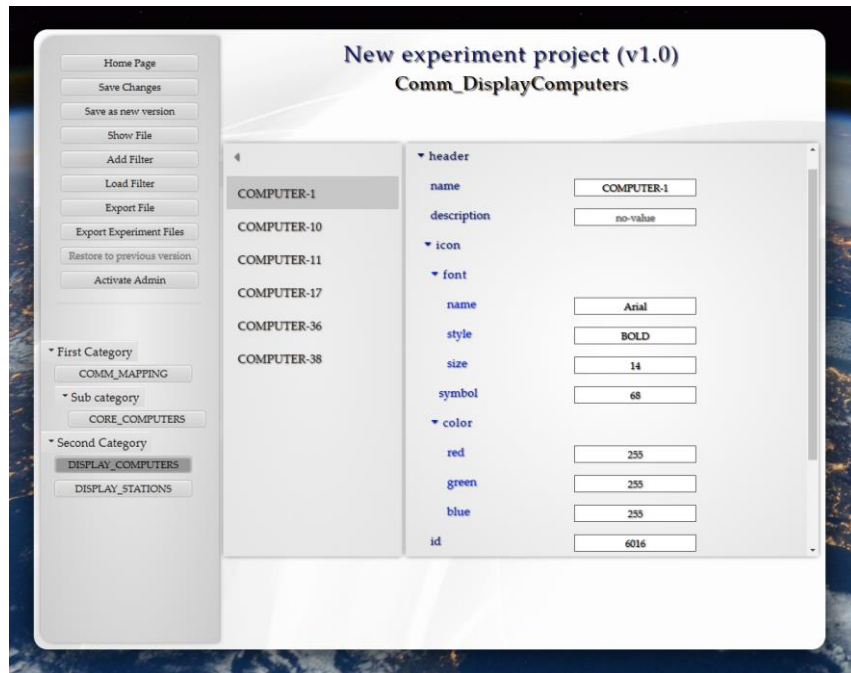
חלון טעינת ניסוי. בחלון זה מוצגים כלל הניסויים השמורים במערכת, לחיצה על אחד מהם תטען אותו.



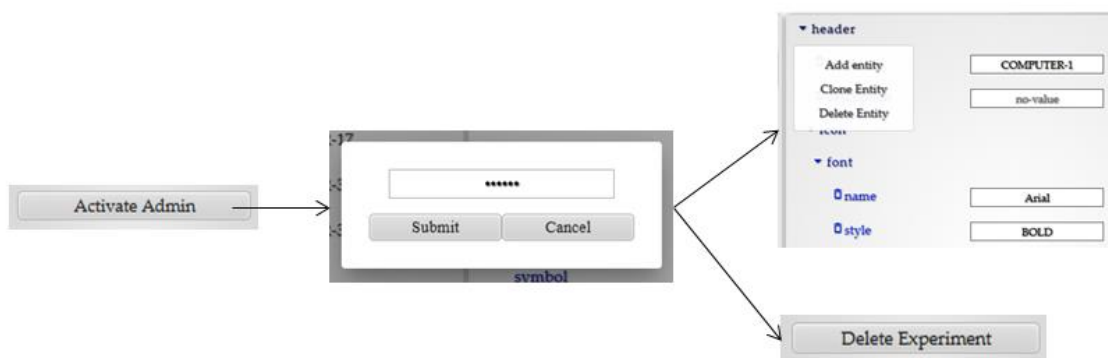
חלון שכפול ניסוי. בחלון זה מוצגים כלל הניסויים השמורים במערכת שאותם ניתן לשכפל. לחיצה על אחד מהם תאפשר מתן שם לניסוי המשובכל ויצירת העותק.



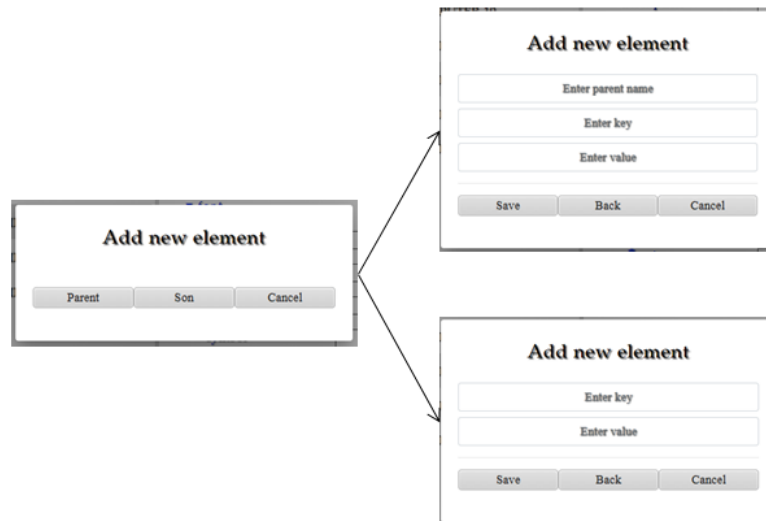
המסך הראשי של המערכת ובו מתבצעת העבודה על קבצי הפרויקט. בחלק השמאלי העליון מופיע תפריט ההגדרות של הניסוי הטעון ובו כלל האפשרויות שהשתמש יכול לבצע. בחלק השמאלי התחתון מופיעות כל הקטגוריות שנוצרו עבור פרוייקט זה ותחתן הקבצים ששויכו אליהן לחיצה על שם של קטגוריות, ימזער או ירחיב אותה בהתאמה. העמודה האמצעית מציגה את שמן של התגיות הראשיות של הקובץ (כלומר, ברמת עומק 1) לחיצה על אחד מהשמות תציג בחלקה הימני של המערכת את תוכנו.



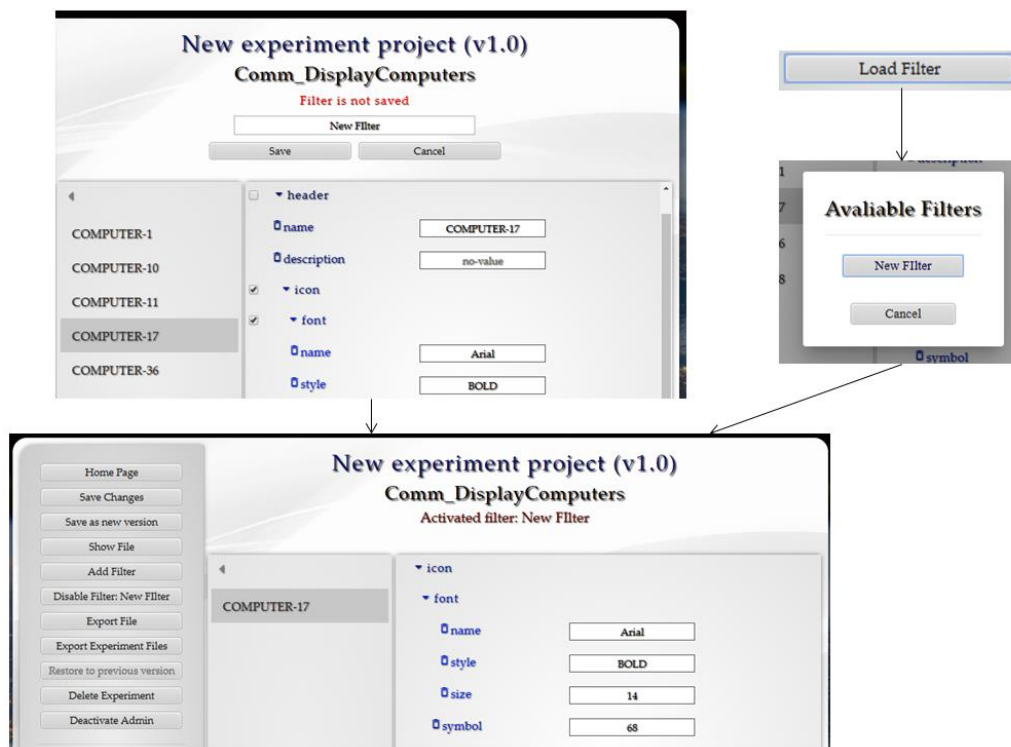
לחיצה בתפריט השמאלי על **Activate Admin** תפעיל מצב ניהול שבו ניתן יהיה לבצע פעולות נוספות: הוספת תגיות, שכפול תגיות, מחיקת תגיות, מחיקת ניסוי.



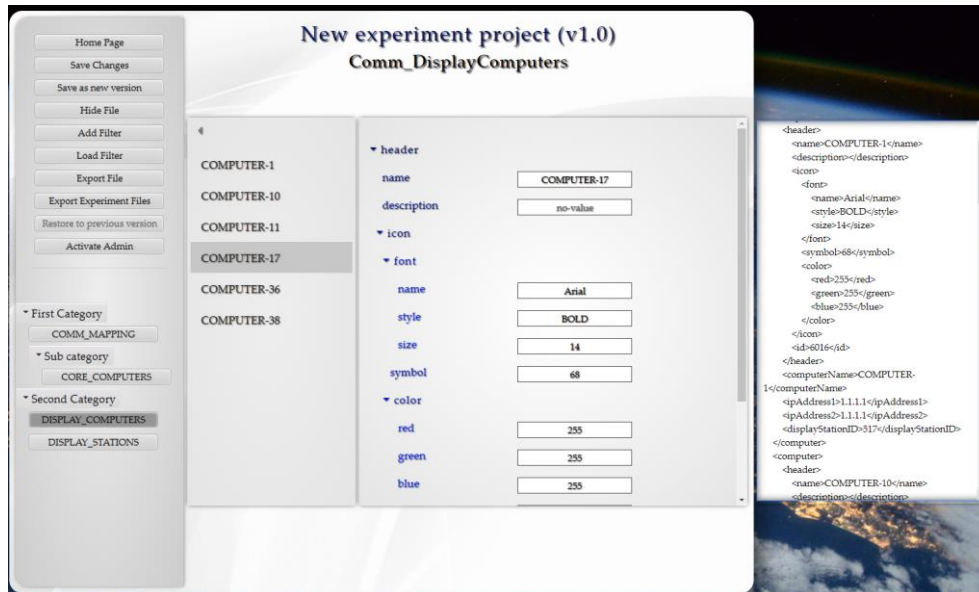
בעת לחיצה על הוספת תגית (**Add entity**), יפתח חלון ובו ניתן יהיה להזין את הפרמטרים הנדרשים. לאחר האישור הוספת התגית תתבצע.



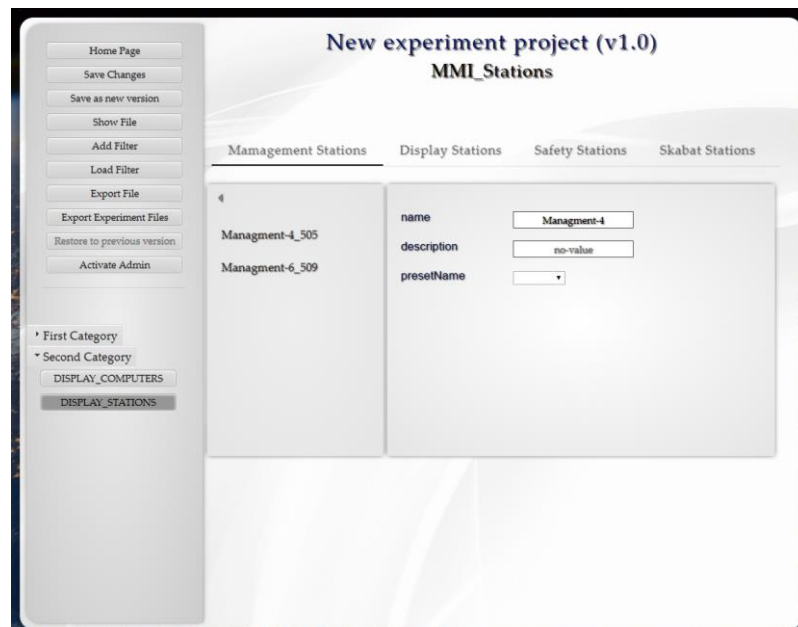
לחיצה על **Add filter** מאפשרת יצירת תצוגה מותאמת על ידי סימון שדות שרוצים לצפות בהם, מתן שם לפילטר ושמירה. מיד לאחר השמירה הפילטר יוצג. במידה ופילטר כבר שמור במערכת, ניתן לטעון אותו על ידי לחיצה על **Load Filter**.



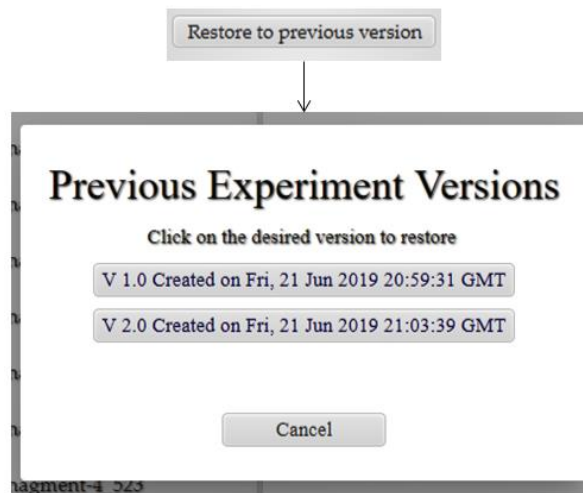
לחיצה על **Show File** תפתח את תוכן קובץ ה-XML בפורמט המקורי. ניתן לעקוב אחרי השינויים שמתבצעים בזמן אמת בקובץ וכמו כן לצבע שינויים על גבי הטופס עצמו ולצפות בשינוי גם בטופס (בעצם זו מערכת דו כיוונית).



המערכת תומכת בהצגה של קבצים בתצוגה מותאמת שפותחה במיוחד עבורם. כחלק מהמשך פיתוח עתידי, ניתן יהיה לייצר תצוגה מותאמת עבור כל קובץ XML ולאפיין את תצוגתו בהתאם. ניתן לראות בתמונה המופיע בהמשך שתצוגת הקובץ DISPLAY_STATIONS מחולקת ל 2 רמות, הרמה הראשונה מוצגת בטאבים האופקיים והרמה השנייה מוצגת ברשימה האנכית משמאל. בכך נוצרת תאימות טובה יותר עבור הקובץ הספציפי הנ"ל. בנוסף, מופעל פילטר קבוע שמציג רק חלק מהשדות, על פי דרישה.



במידה וקיימות גרסאות קודמות של הניסוי, לחיצה על Restore to previous version תעלה חלון ובו יוצגו הגרסאות הקודמות שנשמרו עבור אותו ניסוי. לחיצה על אחת מהן תוביל לשחזור הפרויקט לנקודת הזמן הרשומה.



Abstract

Experiments and missile launches of the Israeli security forces are supervised and controlled via systems that use large amount of parameter files that define all relevant activities that occur during those experiments, including information that is passed via sensors and stations that activate processes or observe them in real time.

Control of these files is necessary for the existence of those activities and producing the most of them.

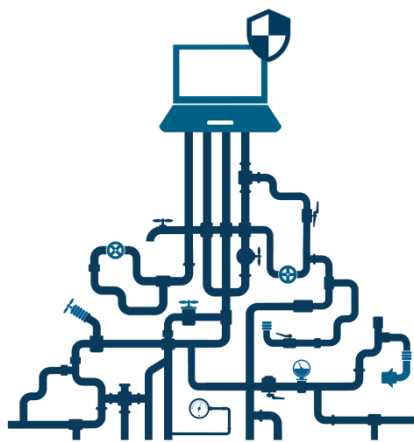
Usually, the structure of these files is complicated for understanding and editing in a direct manner. From this reason, a system that is capable of simplifying the data for the user is needed so that he would be able to take control of the large amount of information.

In addition, there is a need to manage parameter files in an efficient way so that It would be possible to restore files to their previous versions.

This report records the development of a system that gives an answer to the described problem.

Software Engineering Department

Web Configurations Dashboard



By

Alex Levenzon

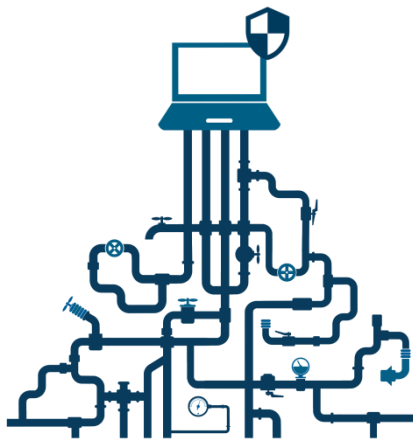
Alon Schwartz

Academic Supervisor

Dr. Reuven Yagel

Software Engineering Department

Web Configurations Dashboard



By

Alex Levenzon

Alon Schwartz

June 2019

Sivan 5779