

מסמך עיצוב

מסמך זה מתאר את עיצוב המערכת שנקבע לפי מסמך האיפיון.

המסמך בנוי משלושה חלקים:

- טכנולוגיות
- תיאור הפונקציונליות ב- backend ומבנה בסיס הנתונים
- תיאור המסכים השונים במערכת

טכנולוגיות –

המערכת תפותח על גבי ASP.net core web api, כך שהיא כוללת backend שהוא בעיקרו controller המגיבים לבקשות http המניעות תהליכים במערכת. יהיה ניתן להתקין את התוצרים במערכות הפעלה מוכרות כגון Windows ו- linux.

המערכת תכיל קבצים סטטיים, שיוגשו ללקוח המבקש והם ממשקי המשתמש והלוגיקה והעיצוב בצד הלקוח באמצעות קבצי html, javascript ו- css.

טכנולוגיות צד הלקוח בלתי תלויות בטכנולוגיות צד השרת, כך שכל מפתח html, javascript ו- css יוכל להשתתף בפיתוח, גם אם אינו יודע מהו dotnet ובפרט ASP.net. זוהי טכנולוגיה אוניברסלית, בעלת שימוש נרחב בעולם וקלה לשימוש.

צד השרת נגיש באמצעות בקשות http כאמור אשר לא מכילות מאפיין מיוחד ל- asp.net כך שאם ה- backend יוחלף בכל טכנולוגיה אחרת כגון PHP, Apache tomcat, node.js, צד הלקוח ימשיך לתפקד כנדרש ללא צורך בשינוי מיוחד.

בסיס הנתונים יהיה Microsoft SQL Server Express אשר ה- backend ייגש אליו לאחסון ושליפת נתונים.

תיאור הפונקציונליות ומבנה בסיס הנתונים –

בבסיס הנתונים שתי טבלאות קבועות מרכזיות. שאר הטבלאות נוצרות באופן דינמי ע"י המשתמשים של המערכת.

טבלת ארגונים – הארגון הוא הלקוח של המערכת. לכל ארגון מזהה ארגון ייחודי (סטרינג).

טבלת משתמשים – לכל משתמש, שם משתמש, מזהה ארגון אליו הוא משתייך, סיסמה, שעת התחברות אחרונה (לצורך תיעוד) ותפקיד/רמת הרשאה (מספר שלם)

שאר הטבלאות נוצרות באופן דינמי בזמן ריצה ע"י משתמשי העל. לכל טבלה כזו שנוצרת, עמודת id שהיא מפתח ראשי, מספר רץ. שמות הטבלאות הם בעלי משמעות מיוחדת על מנת להפריד בין הארגונים.

כל טבלה שכזו תהיה בעלת שם מהתבנית: t_[organization id]_[table name].

בצד השרת, המחלקות ב- C# הן:

ה- Controller:

- DataController – עוסק בהזנה ועריכה נתונים
 - פונקציה getAllData מחזירה את כל המידע השמור בכל הטבלאות של הארגון.

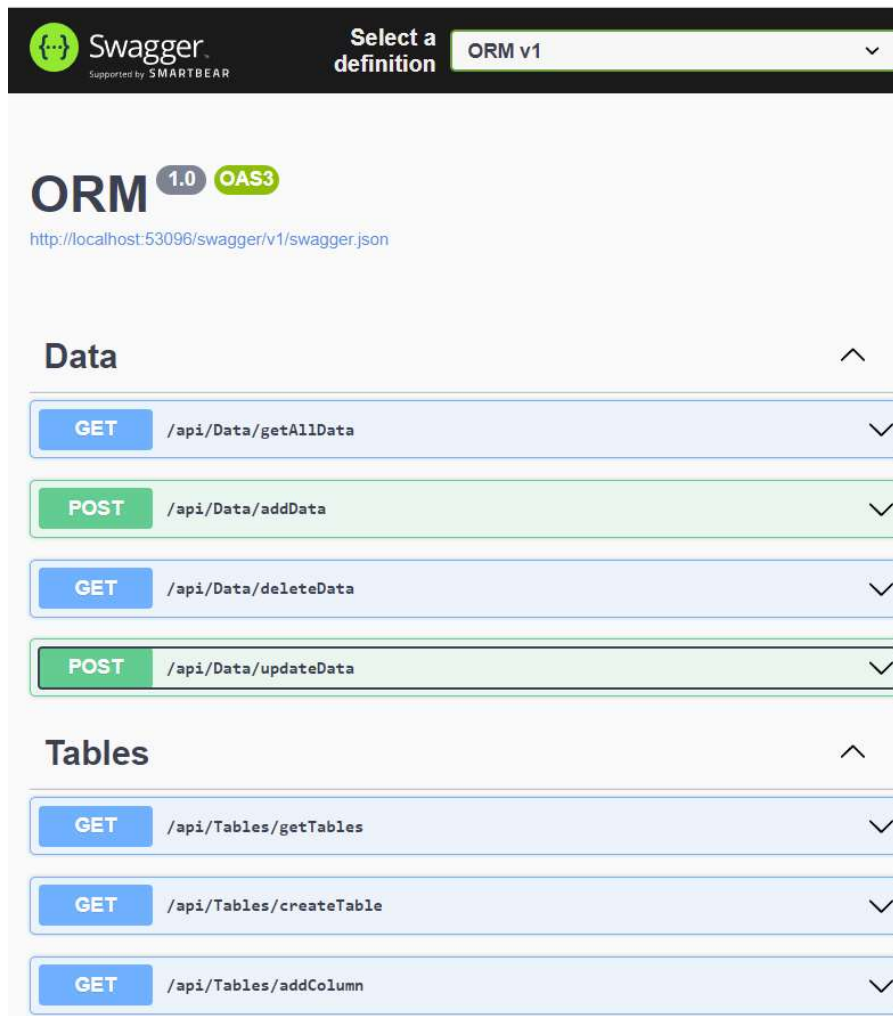
- פונקציה addData מאפשרת הוספה של רשומה לטבלה מסויימת. זוהי בקשת POST אשר ערכי העמודות מועברים ב- body באמצעות dictionary שכל key הוא שם העמודה וה- value שלו זה ערך העמודה.
- פונקציה deleteData מאפשרת למחוק רשומה מסויימת מטבלה, לפי שם הטבלה והמזהה הייחודי שלה
- פונקציה updateData מאפשרת עדכון רשומה. דומה במבנה ל- addData אלא שמצופה שב- dictionary שבו body של הבקשה, יהיה key שהוא id.
- UsersController – עוסק בניהול משתמשים
 - פונקציית login אשר מקבלת כפרמטר username, password, organizationId ומחזירה bool אשר מאשר שפרטי ההתחברות נכונים
 - פונקציית signup אשר מקבלת פרמטרים זהים ל- login ומוסיפה גם ארגון חדש וגם משתמש חדש שהוא יהיה מנהל הארגון. מחזירה true אם הפעולה הצליחה (היא תיכשל אם כבר קיים כזה משתמש בארגון הזה)
 - פונקציית getUsers אשר מחזירה רשימה של כל המשתמשים בארגון, לרבות הפרטים שלהם כגון רמת הרשאה.
 - פונקציית deleteUser – אשר מאפשרת למחוק משתמש
 - פונקציית addUser – מנהל הארגון יכול להוסיף משתמשים תחת הארגון.
 - פונקציית changePassword – מאפשרת למשתמש לשנות את הסיסמה של עצמו.
 - פונקציית resetPassword – מאפשרת למנהל הארגון לאפס סיסמה למשתמש אחר בארגון ששכח את סיסמתו לסימט ברירת מחדל 12345 שאותה המשתמש ישנה ע"י changePassword
 - פונקציית changeRole – מאפשרת לעדכן למשתמש את ה- role שלו.
- TablesController – עוסק בניהול מבנה בסיס הנתונים
 - פונקציית getTables – מחזירה רשימה של כל הטבלאות בארגון לרבות העמודות שלהן.
 - פונקציה createTable – יוצרת טבלה חדשה בארגון, בעלת עמודה אחת של id.
 - פונקציית addColumn – מאפשרת הוספת עמודה לטבלה קיימת.
 - פונקציית dropColumn – מאפשרת הסרת עמודה קיימת מטבלה קיימת. אם עמודה זו מכילה מפתח זר, יש להסירו ראשית.
 - פונקציית dropTable – מסירה טבלה שלמה, את הנתונים ואת קיומה. מתאפשר רק אם אין טבלה אחרת אשר מתייחסת לטבלה זו.
 - פונקציית addForeignKey – מאפשרת הוספת מפתח זר לעמודה קיימת תוך ציון הטבלה אליה תהיה הפנייה.
 - פונקציית dropForeignKey – מאפשרת הסרה של מפתח זר מעמודה קיימת

כמעט כל הפונקציות ב- controller השונים, המקבלות בקשות http מצפות לשלושה פרמטרים קבועים בנוסף לשאר הפרמטרים, והם שם משתמש, ארגון וסיסמה וזאת על מנת לזהות את המשתמש. אפשר היה להשתמש במנגנון של טוקן, כלומר בהתחברות מוקצה טוקן למשתמש ושאר הפעולות הן באמצעות אותו הטוקן.

כל הפונקציות הנ"ל הן פונקציות GET למעט אילו שהן POST ונכתב במפורש שהן כאילו. למעט הרשמה למערכת, לכל הפונקציה יש להעביר (סטרינגים, ב- QueryString):

username, password, organizationId

באמצעות Swagger יוצג ה-API בפירוט רב. זהו כלי שבזמן הרצת האפליקציה בונה מסכים המציגים את הפונקציונליות השונה וה-API הנחשף באופן אחיד הקריא גם לאדם וגם למכונה:



The image shows the Swagger UI for an API named 'ORM v1'. The top bar includes the Swagger logo, the text 'Supported by SMARTBEAR', and a dropdown menu to 'Select a definition' with 'ORM v1' selected. Below the header, the API name 'ORM' is displayed with version '1.0' and 'OAS3' tags. The URL 'http://localhost:53096/swagger/v1/swagger.json' is shown. The main content is divided into two sections: 'Data' and 'Tables'. The 'Data' section contains four endpoints: GET /api/Data/getAllData, POST /api/Data/addData, GET /api/Data/deleteData, and POST /api/Data/updateData. The 'Tables' section contains three endpoints: GET /api/Tables/getTables, GET /api/Tables/createTable, and GET /api/Tables/addColumn. Each endpoint is represented by a colored button (blue for GET, green for POST) and a dropdown arrow.

מחלקות מרכזיות נוספות ב- backend:

- User – מתאר משתמש במערכת
 - שם משתמש
 - תפקיד Role
- Table – מתאר מבנה טבלה במערכת, על העמודות שלה
 - שם הטבלה
 - רשימת העמודות של הטבלה
- DB – זוהי מחלקת עזר (בעלת פונקציות סטטיות) המבצעת הפשטה של הגישה לבסיס הנתונים. ה- Controller השונים יישמשו במחלקה זו כדי לגשת לבסיס הנתונים.
 - שומר את ה- Connection String שנשלף בעליית המערכת.
 - פונקציה לשליפת נתונים
 - פונקציה לעדכון נתונים
- Column – אובייקט ממחלקה זו מייצג עמודה בטבלה כלשהי בבסיס הנתונים
 - שם העמודה
 - מיקום העמודה ברשימה
 - טיפוס הנתונים
 - ערך ברירת מחדל
 - האם מותר null ?
 - אורך מירבי של מחרוזת

○ טבלת מפתח זר

עפ"י המקובל ב- ASP.net core, קובץ appsettings.json יכיל את פרטי החיבור לבסיס הנתונים.

בצד הלקוח, הקבצים הבאים:

- index.html – למעשה למערכת קובץ html יחיד. מעבר בין מסכים הוא פנימי ולא מצריך מעבר בין קבצי html.
- style.css – קובץ עיצוב ממשק המשתמש. עיצוב המערכת הוא מינימלי ביותר וניתן להוסיף לקובץ זה עיצוב נרחב יותר.
- script.js קובץ אשר מכיל פונקציות עזר, פונקציות איתחול, ופונקציות אשר מעבירות בין שלושת המסכים של המערכת
- קובץ users.js המכיל את כל הלוגיקה בקליינט העוסקת במשתמשים.
- קובץ tables.js המכיל את כל הלוגיקה בקליינט העוסקת בעיצוב מבנה בסיס הנתונים
- קובץ data.js המכיל את הלוגיקה בקליינט העוסקת בהצגה ועריכה של נתונים.

תיאור המסכים השונים במערכת

(משמש גם כמדריך למשתמש)

מסך התחברות/הזדהות:

login or signup

username:

password:

organization:

יש להזין שם משתמש, סיסמה וארגון. מומלץ לא להשתמש בשמות המכילים תווים שאינם אלפא נומריים. יש ללחוץ על login אם כבר נרשמת או signup אם זו הרשמה של ארגון חדש.

מסך ניהול משתמשים:

welcome to Information System Wizard

users design DB data entry

users management

users: elad ▼ role: 1 reload delete user reset password

change role: set role

new user: add user

(users are added with default password of 12345 and they can change it anytime)

change my password to: set my password

ניתן לראות כי בכל שלושת מסכי המערכת העיקריים, החלק העליון קבוע והוא מאפשר מעבר בין שלושת המסכים השונים.

בגלל שרוב הכפתורים במערכת גורמים להוצאת בקשת http, תהליך שעלול להיות איטי בהתאם למהירות האינטרנט של המחשב, חלק מהכפתורים ושדות הקלט ננעלים בזמן הבקשה, שמבוצעת באופן אסינכרוני, וזאת כדי לבטא למשתמש שיש תהליך ברגע ויש להמתין לסימו.

מסך ניהול מבנה בסיס הנתונים:

welcome to Information System Wizard

users design DB data entry

design DB

tables: mytable ▼ reload delete table: DROP this table

create new table: create table

add column: name: data type: text ▼ is nullable? ☐ default value: add new column

columns of selected table: id ▼ drop this column

datatype: integer. ordinal position: 1. is nullable? no. character length 0. foreign key: DROP this foreign key

add foreign key: mytable ▼ add foreign key

(first select a column. the foreign key is always to the 'id' column.)

במסך זה יש תיבת בחירה של טבלה מתוך הטבלאות הקיימות. חלקים שונים במסך מתייחסים לטבלה הנבחרת הזו. כלומר, כפתור ה- DROP this table מתייחס לטבלה הנבחרת בתיבת בחירה זו, אבל גם הוספת עמודה add new column היא לטבלה זו. באותו אופן, יש תיבת בחירה של עמודה וחלקים שונים מתייחסים לערך הנבחר, גם drop this column וגם DROP this foreign key מתייחסת לעמודה הנבחרת. בדוגמא לעיל, הכפתור DROP this foreign key מושבת כי לעמודה זו אין מפתח זר.

מומלץ להשתמש בשמות בתווים אלפא נומריים בלבד (השרת לא בודק תווים לא חוקיים אבל יניב שגיאה אם לא יוכל לבצע את הפעולה).

מסך הזנה ועריכת נתונים:

welcome to Information System Wizard

users design DB data entry

edit and view data

table: mytable2 ▼ reload

id	col2	foreigntest		
1	vfgfrgvd	2	delete	edit
2	hyhgtg	3	delete	edit

add new data:

id

col2

foreigntest ▼

save

לאחר בחירת טבלה מתיבת הבחירה, תוצג הטבלה המתאימה במסך. לחיצה על delete בשורה מסויימת, תמחק את השורה הזו. לחיצה על edit תגרום לטופס שמתחת לטבלה להתאכלס בערכים המתאימים, ואז ניתן לשנות ערכים וללחוץ על save. לפני כן, הטופס נמצא במצב של הוספת רשומה חדשה. ניתן להבחין בין המצבים ע"י השדה id. הוא מופיע במסך readonly ואם יש ערך לשדה זה אז אנחנו במצב עריכה (חסרה אפשרות לצאת ממצב עריכה ולחזור למצב הוספה שלא ע"י שמירה, היינו cancel, כרגע ניתן לעבור לטבלה אחרת ואז לחזור לזו).

כפתורים רבים במערכת, גורמים לשרשרת אירועים של רענון ועדכון המסך.