**EZSchool**

בתחילת הפרויקט הגדרנו 33 דרישות , בהמשך העבודה הגדרנו 12 דרישות נוספות על מנת לשפר את המערכת ולייצר סביבת עבודה יעילה ,דינאמית ונוחה לשימוש בזמן משבר הקורונה.  
הדרישות ממושו לפי התיעדוף שהוגדר . חלק מהסיפורים בוצעו באופן אישי וחלקם בוצעו תוך שימוש בmob programing משום שהיו מורכבים .

בפועל פותחו סה"כ 44 דרישות עסקיות מתוך 45 דרישות שתוכננו, הדרישות מחולקות לפי חבר הצוות שמימש אותן – ממוספרות לפי המספרים שהוגדרו במייסטרטאסק.

**קישור GitHub:**  
<https://github.com/yovelaloni/EZSchool/tree/master>

**קישור לתקייה בדרייב:**

<https://drive.google.com/drive/folders/1mOvRsPTZqJGjcB-ncomgjIUtotW9lzFN?usp=sharing>

**שם קבוצה בmeisterTask:**2020\_21\_project#group\_18

**קישור לmeisterTask:**

https://www.meistertask.com/app/project/t2Hr3Ev2/2020-21-project-group-18

**פירוט דרישות:**

**יובל:**

4. כמזכירה אוכל למלא הצהרת בריאות על מנת לאפשר הגעה פיזית לבית הספר במידת הצורך"add health yovel"

5. כמזכירה תינתן לי האפשרות לעדכן מערכת שעות של מורה-"schedule yovel"

6. כמזכירה תינתן לי האפשרות לעדכן מערכת שעות של תלמיד-"schedule yovel"

11. כמזכירה תינתן לי האפשרות לגבות תשלום עבור שכר לימוד של התלמידים –"payment yovel"

18. כמורה תינתן לי האפשרות למלא הצהרת בריאות על מנת לאפשר הגעה פיזית לבית הספר במידת הצורך-"add health yovel"

19. כמורה תהיה לי גישה למערכת שעות שלי על מנת לנהל לו"ז בצורה טובה-"schedule yovel"

22. כמורה אוכל ליצור ולגשת לקישורי זום על מנת להעביר שיעורים מקוונים - –"zoom links students and teacher"

24. כתלמיד תהיה לי גישה למערכת השעות שלי על מנת שאוכל לדעת מתי השיעורים שלי מתקיימים ואם היו שינויים-"schedule yovel"

27. כתלמיד תינתן לי גישה לקישורי הזום של השעורים על מנת שאוכל להיכנס לשיעור –"zoom links students and teacher"

29. כתלמיד תיתנן לי האפשרות לשלם את שכר הלימוד לבית הספר על מנת שלא יווצר לי חוב–"payment yovel"

30. כתלמיד תינתן לי האפשרות למלא הצהרת בריאות על מנת לאפשר הגעה פיזית לבית הספר-"add health yovel"

45. כמזכירה אוכל להנפיק דוח הצהרות בריאות עדכני להיום על מנת לדעת מי יכול להיכנס לבית הספר במידת הצורך – "add health report"

**רפאל**:

8. כמזכירה תינתן לי האפשרות לנהל מלאי של ציוד משרדי- "added shop student + teacher and payment method"

16. כמורה אוכל להעלות מטלות ושיעורי בית על מנת שתלמידים יוכלו לתרגל את החומר הנלמד-"upload/download matalot"

17.כמורה אוכל לתת משוב למטלות של התלמידים על מנת שיכולו להשתפר-"feedback teacher"

21. כמורה אוכל להעלות חומרי לימוד על מנת שהתלמידים יוכלו לחזור על החומר–"teacher uploads+student down/upld"

23. כתלמיד אוכל להיכנס למערכת על מנת לגשת לחומרי הלימוד שלי - hackaton

26. כתלמיד אוכל להעלות מטלות ושעורי בית על מנת שאוכל לקבל ציון ומשוב עליהם

28.כתלמיד תינתן לי האפשרות לצפות בחומרי הלימוד שהועלו על ידי המורים על מנת שאוכל להיות מעודכן –"teacher uploads+student down/upld"

38. כתלמיד אוכל להפיק דוח ציונים על מנת להיות במעקב על ההישגיים הלימודיים שלי-"GradesReport finished"

39. כמורה אוכל להפיק דוח ציונים על מנת להיות במעקב על ההישגיים של התלמידים-"GradesReport finished"

40. כמזכירה אוכל להפעיל דוח של מלאי ציוד זמין על מנת שאוכל להיות במעקב-"Attendance and inventory report"

44. כמורה אוכל לגשת לדוח הרכישות על מנת לבצע מעקב על הזמנות ציוד שעשיתי-"Attendance and inventory report"

**טל:**

7. כמזכירה תינתן לי האפשרות לגבות תשלום עבור הזמנת ציוד משרד-"payment yovel"

10. כמזכירה אוכל לגשת לפורום על מנת לעדכן את שאר המשתמשים בהודעות- hackaton

12. כמורה אוכל להיכנס למערכת על מנת לבצע פעולות במסגרת תפקידי -hackaton

14. כמורה תינתן לי האפשרות להזמין ציוד משרדי ללימודים על מנת לבצע את מטלותיי כמורה

15. כמורה אוכל לסמן את נוכחות התלמידים בשיעור על מנת לבצע מעקב נוכחות-"Attendance and inventory report"

20. כמורה תהיה לי גישה לפורום על מנת לתקשר עם התלמידים- hackaton

21.כתלמיד אוכל לגשת לפורום על מנת לתקשר עם צוות בית הספר- hackaton

34. כמורה תינתן לי האפשרות להוציא דוח חיסורים לפי כיתות על מנת שאוכל להיות במעקב- "added attendance"

42. כתלמיד אוכל לגשת לדוח הרכישות שלי על מנת לבצע מעקב על הזמנות ציוד שביצעתי– "Attendance and inventory report"

43. כמזכירה אוכל לגשת לדוח ניהול יתרות תשלום שכר הלימוד על מנת להיות במעקב אחרי תשלומי שכר הלימוד –"report balance"

**ויקטוריה:**

1.כמזכירה אוכל להכנס למערכת כדי לבצע פעולות במסגרת תפקידי–"hackaton"

2.מזכירה אוכל למלא דוח שעות על מנת לבצע מעקב אחר שעות העבודה שלי – "work clock"

9. כמזכירה אוכל לנהל סקר מורים על מנת לדעת את שביעות רצונם של התלמידים-"Survey update"

13.כמורה אוכל למלא דוח שעות על מנת לבצע מעקב אחרי שעות העבודה שלי– "work clock"

32. כתלמיד תינתן לי האפשרות לגשת למאגר משחקים לימודיים שהועלו על ידי המורים-"Games update"

33. כמורה תינתן לי האפשרות להוסיף קישורים לדף המשחקים-"Games update"

35. כתלמיד תינתן לי האפשרות למלא סקר הוראה על מנת שאוכל להביע את דעתי-"Survey update"

36. כמזכירה אוכל להנפיק דוחות של שעות העבודה שלי על מנת שאוכל להיות במעקב-– "work clock"

7. כמורה אוכל להנפיק דוחות של שעות העבודה שלי על מנת שאוכל להיות במעקב– "work clock"

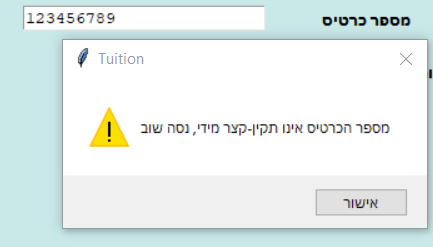
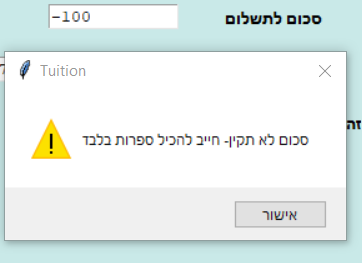
41. כתלמיד אוכל לגשת דוח נוכחות על מנת לבצע מעקב אחר החיסורים שלי – "Attendance and inventory report"

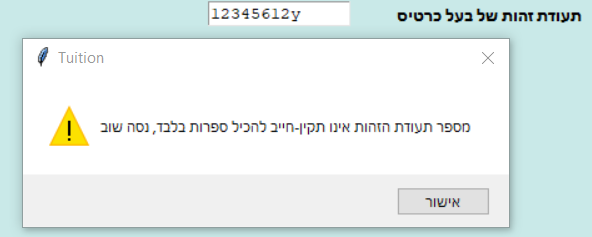
**דו"ח בדיקות:**

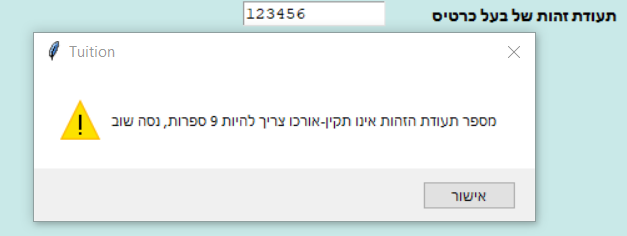
בוצעו המון בדיקות ידניות במהלך כתיבת הקוד .   
בנוסף ,בכל דף בו מתבצע קליטת נתונים מתבצעות בדיקות של תקינות הקלט ובהתאם לכך מוצגת הודעות למשתמש .

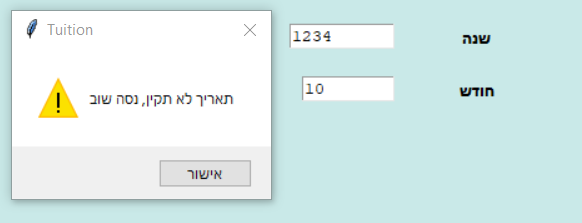
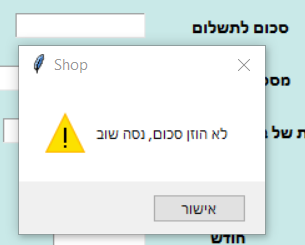
לדוגמא בדף של תשלום :

תתבצע בדיקה לתקינות כל אחד מהערכים ותופיע אזהרה בהתאם :  
סכום – חיובי , מכיל רק ספרות  
מס' כרטיס אשראי – אורכו מינימום 16 ומכיל רק ספרות  
תעודת זהות- אורכה 9 ומכילה רק ספרות  
תאריך – עדכני ועתידי



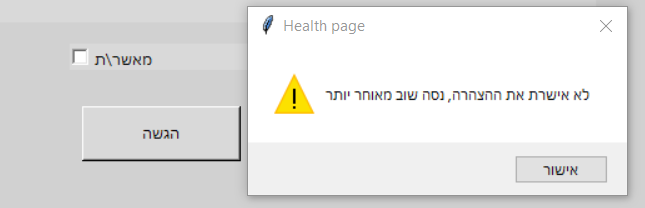


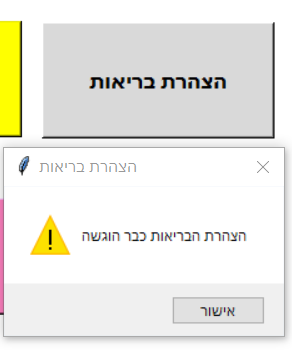




בדף של הצהרת בריאות :

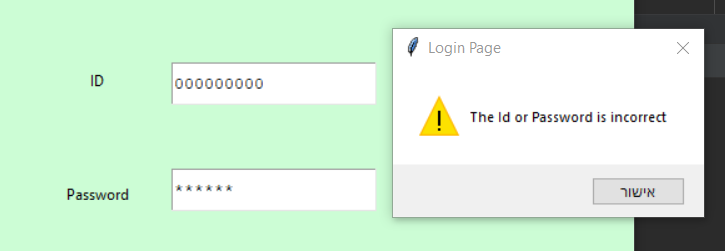
אם היה נסיון להגיש את הצהרת הבריאות ללא אישור תופיע אזהרה



אם הוגשה כבר הצהרה באותו יום :

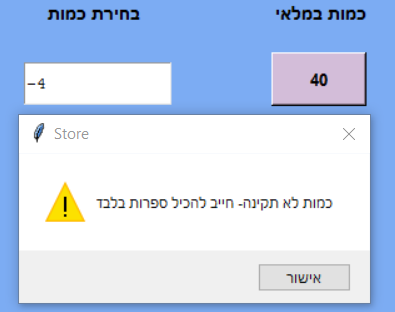
בדף של login:

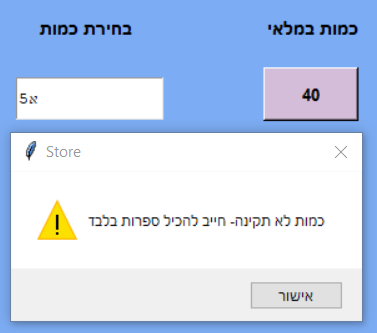
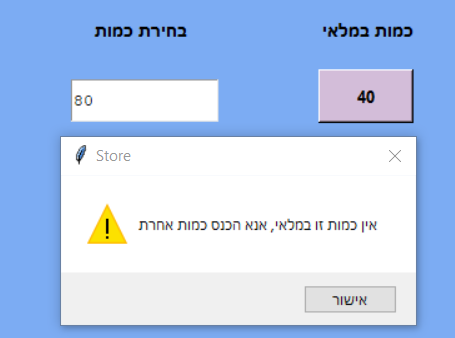
מתבצעת בדיקה אם תעודת הזהות נמצאת בDB , אם היא נמצאת תתבצע בדיקה לססמא אם תואמת למס' תעודת הזהות .  
אם תעודת הזהות לא מופיעה או לא תואמת לססמא – תופיע אזהרה



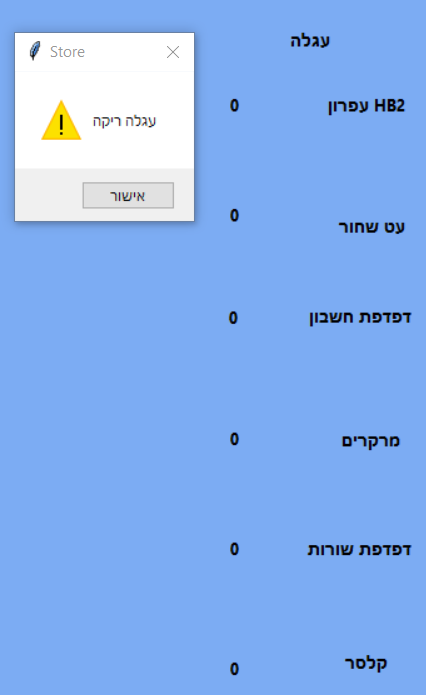
בדף של רכישת ציוד:

מתבצעת בדיקה לתקינות של מס' המוצרים שרוצים להכניס לסל בהתחשב בכמות הנמצאת במלאי, ובנוסף בדיקה לקלט תקין של כמות – כלומר מורכב רק מספרות ,חיובי וכו' .

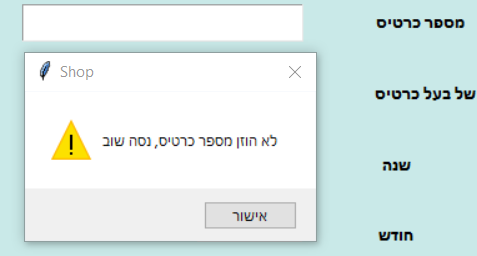
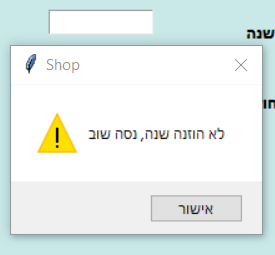


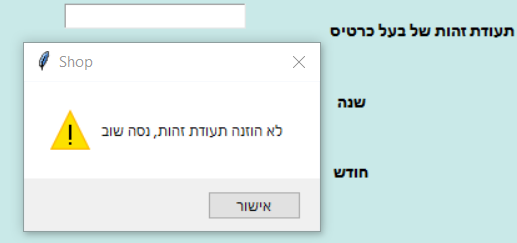
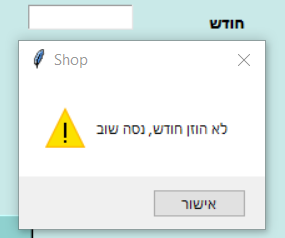


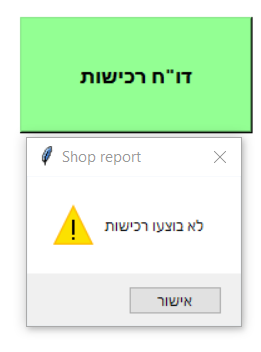
במידה ומתבצע ניסיון לתשלום על עגלה ריקה :



בתשלום לרכישת ציוד :



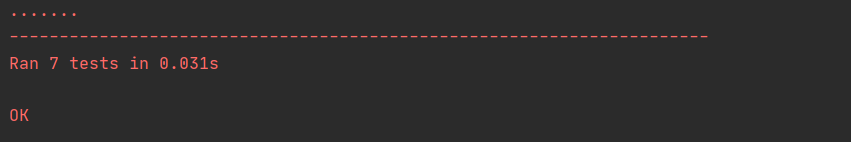




בהנפקת דוח רכישות : במידה ומתבצע ניסיון להנפקת הדוח לפני שבוצעו רכישות

בנוסף , התבצעו בדיקות יחידה על פונקציות נוספות ועל מאגרי הDB – על תקינותם ועדכונם .

import unittest  
import sys  
import pymongo  
import tkinter  
sys.path.append('..')  
from data import db\_operations  
from data import user\_db\_init, inventory\_db\_init,game\_links\_db\_init,Schedule\_db\_init,ScheduleTeacher\_db\_init,seker\_db\_init  
user\_db\_init()  
inventory\_db\_init()  
game\_links\_db\_init()  
Schedule\_db\_init()  
ScheduleTeacher\_db\_init()  
import ScheForOneClassSecretary  
client = pymongo.MongoClient()  
mydb = client['EZSchooldb']  
  
  
class MyTestCase(unittest.TestCase):  
 def test\_isinUsers(self):  
 *"""  
 checks if the user is in the DB by id  
 """* global mydb  
 # 123456789 is in DB  
 userobj = mydb['Users'].find\_one({'id':'123456789'})  
 self.assertIsNotNone(userobj,"sould not be NONE")  
  
 # 000000000 is not in DB  
 userobj = mydb['Users'].find\_one({'id':'000000000'})  
 self.assertIsNone(userobj,"sould be NONE")  
  
 def test\_checkUpdateUser(self):  
 *"""  
 checks if filed update is working  
 """* #find one user  
 global mydb  
 userobj = mydb['Users'].find\_one({'id':'123456789'})  
  
 #current password:1234  
 self.assertEqual("1234",userobj['password'],"should be equal")  
  
 #update new password:123456  
 newpass ="123456"  
 mydb['Users'].update\_one({'id':userobj['id']}, {'$set': {'password':newpass}})  
 userobj = mydb['Users'].find\_one({'id':'123456789'})  
 self.assertEqual(newpass,userobj['password'],"should be equal")  
  
 #update old password: 1234  
 mydb['Users'].update\_one({'id':userobj['id']}, {'$set': {'password':"1234"}})  
  
 def test\_getUser(self):  
 *"""  
 checks if the function returns the correct user  
 """* f = open("Current\_user.txt", "w")  
 txt="319122842\nYovel Aloni"  
 f.write(txt)  
 f.close()  
  
 global mydb  
 userobj = mydb['Users'].find\_one({'id':'319122842'})  
 userobj1=db\_operations.getUser()  
 self.assertEqual(userobj1['id'],userobj['id'],"should be equal")  
  
 def test\_getSuject\_and\_setSuject(self):  
 *"""  
 checks if the function setSuject updatess the subject and the function getSubject return the correct subject  
 """* subject = "Math"  
 db\_operations.setSubject(subject)  
 self.assertEqual(db\_operations.getSubject(),subject,"should be equal")  
  
 def test\_ScheduleClassesDB(self):  
 *"""  
 the function checks if the DB updates and return correct values  
 """* global mydb  
 userobj = mydb['ScheduleClasses'].find\_one({'name':'class1'})  
 mycol = mydb['ScheduleClasses']  
 nameDay="sunday1"  
  
 #the schedule is not arr  
 arr=["בדיקה","בדיקה","בדיקה","בדיקה","בדיקה"]  
 self.assertNotEqual(userobj[nameDay],arr,"should not be equal")  
  
 #input arr in to schedule  
 myquery = {nameDay: userobj[nameDay]}  
 newval = {"$set": {nameDay: arr}}  
 mycol.update\_one(myquery, newval)  
 userobj = mydb['ScheduleClasses'].find\_one({'name':'class1'})  
 self.assertEqual(userobj[nameDay],arr,"should be equal")  
  
 myquery = {nameDay: userobj[nameDay]}  
 newarr=["תנך","חשבון","חשבון","היסטוריה","היסטוריה"]  
 newval = {"$set": {nameDay: newarr}}  
 mycol.update\_one(myquery, newval)  
 userobj = mydb['ScheduleClasses'].find\_one({'name':'class1'})  
  
 self.assertNotEqual(userobj[nameDay],arr,"should not be equal")  
 self.assertEqual(userobj[nameDay],newarr,"should be equal")  
  
 def test\_ScheduleTeacherDB(self):  
 global mydb  
 userobj = mydb['ScheduleTeacher'].find\_one({'id': '312208523'})  
 mycol = mydb['ScheduleClasses']  
 nameDay = "sunday1"  
  
 # the schedule is not arr  
 arr = ["בדיקה", "בדיקה", "בדיקה", "בדיקה", "בדיקה"]  
 self.assertNotEqual(userobj[nameDay], arr, "should not be equal")  
  
 # input arr in to schedule  
 myquery = {nameDay: userobj[nameDay]}  
 mycol = mydb['ScheduleTeacher']  
 newval = {"$set": {nameDay: arr}}  
 mycol.update\_one(myquery, newval)  
 userobj = mydb['ScheduleTeacher'].find\_one({'id': '312208523'})  
 self.assertEqual(userobj[nameDay], arr, "should be equal")  
  
 myquery = {nameDay: userobj[nameDay]}  
 newarr = ["תנך כיתה ד", "חשבון כיתה ב", "חשבון כיתה ב", "היסטוריה כיתה א", "היסטוריה כיתה א"]  
 newval = {"$set": {nameDay: newarr}}  
 mycol.update\_one(myquery, newval)  
 userobj = mydb['ScheduleTeacher'].find\_one({'id': '312208523'})  
  
 self.assertNotEqual(userobj[nameDay], arr, "should not be equal")  
 self.assertEqual(userobj[nameDay], newarr, "should be equal")  
  
 def test\_inventoryDB(self):  
 global mydb  
 mycol= mydb['Inventory']  
 object=mycol.find\_one({'item\_name': 'עיפרון HB2'})  
 self.assertEqual(object['id'],'001',"should be equal")  
 self.assertEqual(object['price'],2,"should be equal")  
 self.assertEqual(object['qty'],50,"should be equal")  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 unittest.main()



**בית ספר:**מורים  
תלמידים  
מזכירות

**תלמיד:**  
שם מלא  
תעודת זהות  
סיסמא  
יתרת שכר לימוד  
כיתה  
סל קניות  
קבלות  
חיסורים  
מס' תלמיד  
סיווג משתמש(3)  
תאריך הגשת הצהרת בריאות

**מורה:**שם מלא  
תעודת זהות  
סיסמא  
סל קניות  
קבלות  
שעות עבודה  
סיווג משתמש (2)  
תאריך הגשת הצהרת בריאות

**מזכירה:**שם מלא  
תעודת זהות  
סיסמא  
שעות עבודה  
סיווג משתמש(1)תאריך הגשת הצהרת בריאות

