

בית הספר הארצי להנדסאים

שם הסטודנט: בוגדנוב אלסו

תאריך הגשה:

שם המנחה: יבגניה צ'רנומז

מגמה וכיתה: מחשבים

נושא פרויקט הגמר:

מערכת לניהול ומעקב אחר שיעורי בית של תלמידים

150 – 200 מילים, גופן 12

תארו בקצרה ובצורה ממצא את הפרויקט שיצרתם, מה מטרתו? מה הוא מכיל? למי הוא מיועד? אילו בעיות הוא אמור לפתור? מה היתרונות שלו? וכולי' (התקציר הינו נגזרת של המבוא שתכתבו).

יש לשים כאן את הצעת הפרוייקט **המקורית המאושרת** ,
(זאת שחזרה ממזכירות – חתומה) . **אסור** לשנות דבר
בהצעה, מעתיקים אותה בשלמותה!

תוכן עניינים

לפי פרקים ראשיים ומשניים (מלאו והשלימו את המבנה המוצג לפי מה שכתבתם בספר הפרוייקט שלכם). מספור דפים מתחיל מתקציר.
 מספור עמודים מתחיל בתקציר (1). מספר עמוד של מבוא = תקציר + הצעה + תוכן עניינים.
 חובה למחוק שורות מיותרות בתוכן עניינים.

נושא	ע"מ
• מבוא	
• נושאים הקשורים לניתוח מערכת :	
○ דיאגרמת קשר	
○ Dfd	
○ מילון נתונים	
▪ תיאור ישויות	
▪ תיאור מאגרים	
▪ תיאור זרימות	
▪ תיאור תהליכים	
○ מבני נתונים	
○ חלופות שפת מימוש	
○ פירוט בדיקות תוכנה ואופן ביצוען	
○ סיכום + נספחים	
• תיאור אלגוריתמים לפתרון הבעיה	

מבוא

מכיל את מטרת העבודה, ורקע כללי לבחירת נושא העבודה (בין ע"מ אחד לשני ע"מ).
אפשר להוסיף הסבר על הטכנולוגיה הנבחרת.
1-1.5 p

נושאים הקשורים לניתוח מערכת

1.1 טבלת דרישות שהשתנו

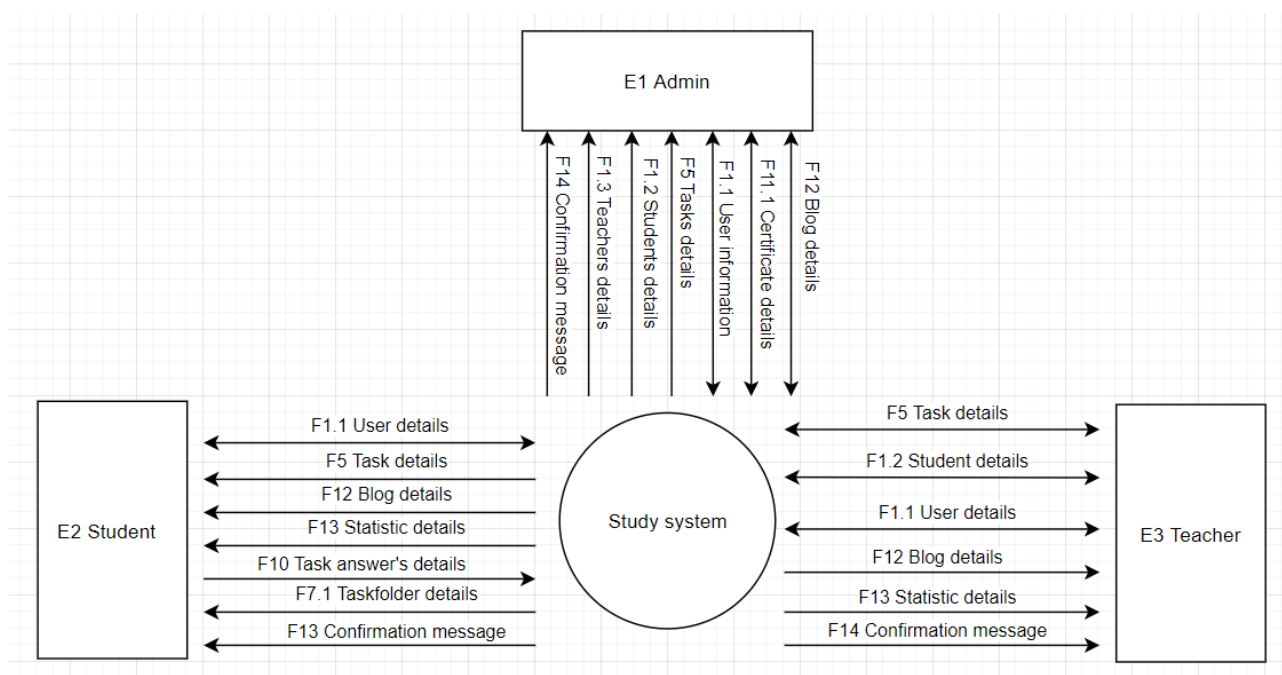
מספר דרישה מקורית	תיאור דרישה מקורית	סיבה לשינוי (דרישה חדשה, נוסח השתנה, דרישה בוטלה)	תיאור חדש של דרישה
	המערכת תכלול אפשרות שחזור סיסמה ושם משתמש (שכחתי סיסמה), שליחת מייל לאיפוס	נוסח השתנה אין צורך לשחזור שם משתמש כי זה מייל והוא יחודי	המערכת תכלול אפשרות שחזור סיסמה דרך שליחת מייל לאיפוס
	המערכת תבדוק ששם המשתמש והמייל לא קיימים כבר במערכת, במקרה וכן תהיה הודעה מתאימה	נוסח השתנה בחרנו ששם משתמש יכול לחפזור את עצמו ורק מייל צריך להיות יחודי	המערכת תבדוק שמייל לא קיים כבר במערכת, במקרה וכן תהיה הודעה מתאימה
	משתמש לא פעיל במשך חצי שנה ימחק.	דרישה בוטלה... (זוגי...)	
	אפשרות הורדה של הסטטיסטיקה (יחיד או קבוצה) בקובץ אקסל בטבלה	דרישה בוטלה	
	המערכת תבצע גיבוי וחשוב של הסטטיסטיקות פעם ביום בשעה 12 בלילה	דרישה בוטלה	
24	לכל המשתמשים יש אפשרות צפייה בכיתות, תיקיות ומשימות וקיימות הרשאות: אדמין, יכול לראות את כל הכיתות, תיקיות, והמשימות מורה, יכול לראות רק את הכיתות, תיקיות, והמשימות שהוא יצר תלמיד, יכול לראות רק את הכיתות, תיקיות, והמשימות שהוא מצורף אליהם.	נוסח השתנה	לכל המשתמשים יש אפשרות צפייה בכיתות, תיקיות ומשימות לפי הרשאות: אדמין: יכול לראות את כל הכיתות, תלמידים, המורים, המשימות ופרטים שלהם מורה: יכול לראות ולתקן רק את הכיתות, תיקיות, קטגוריות, והמשימות שהוא יצר תלמיד: יכול לראות את תיקיות והמשימות שהוא מצורף אליהם. במצב הצלחה נקודות של משימה מתווספות למשתמש (תלמיד) כל משתמש יכול לראות ולבצע את המשימות כלליות
11	המערכת תכלול בממשק המורה אפשרות חלוקה לפי קבוצות (כיתות).		המערכת תכלול בממשק המורה אפשרות חלוקה לפי קבוצות (כיתות) ומקצות
14	המורה יוצר את המשימות ע"י		המורה יוצר את המשימות ע"י

	בחירת תבנית למשימה ומילוי תוכן.		בחירת תבנית למשימה, מילוי תוכן והוספת תכונות של המשימה: משקל, קטגוריה, רמה ותיאור משימה.
9	למורה יש אפשרות למחוק תלמיד מהכיתה במקרה הצורך	נוסח השתנה	למורה יש אפשרות להוסיף תלמיד לכיתה דרך הזנה מייל של תלמיד או הוספת קובץ עם מיילים של תלמידים. גם יש אפשרות למחוק תלמיד מהכיתה במקרה הצורך
	משימות נוספות		
	blog		

דרישה יתפספה עקב ...

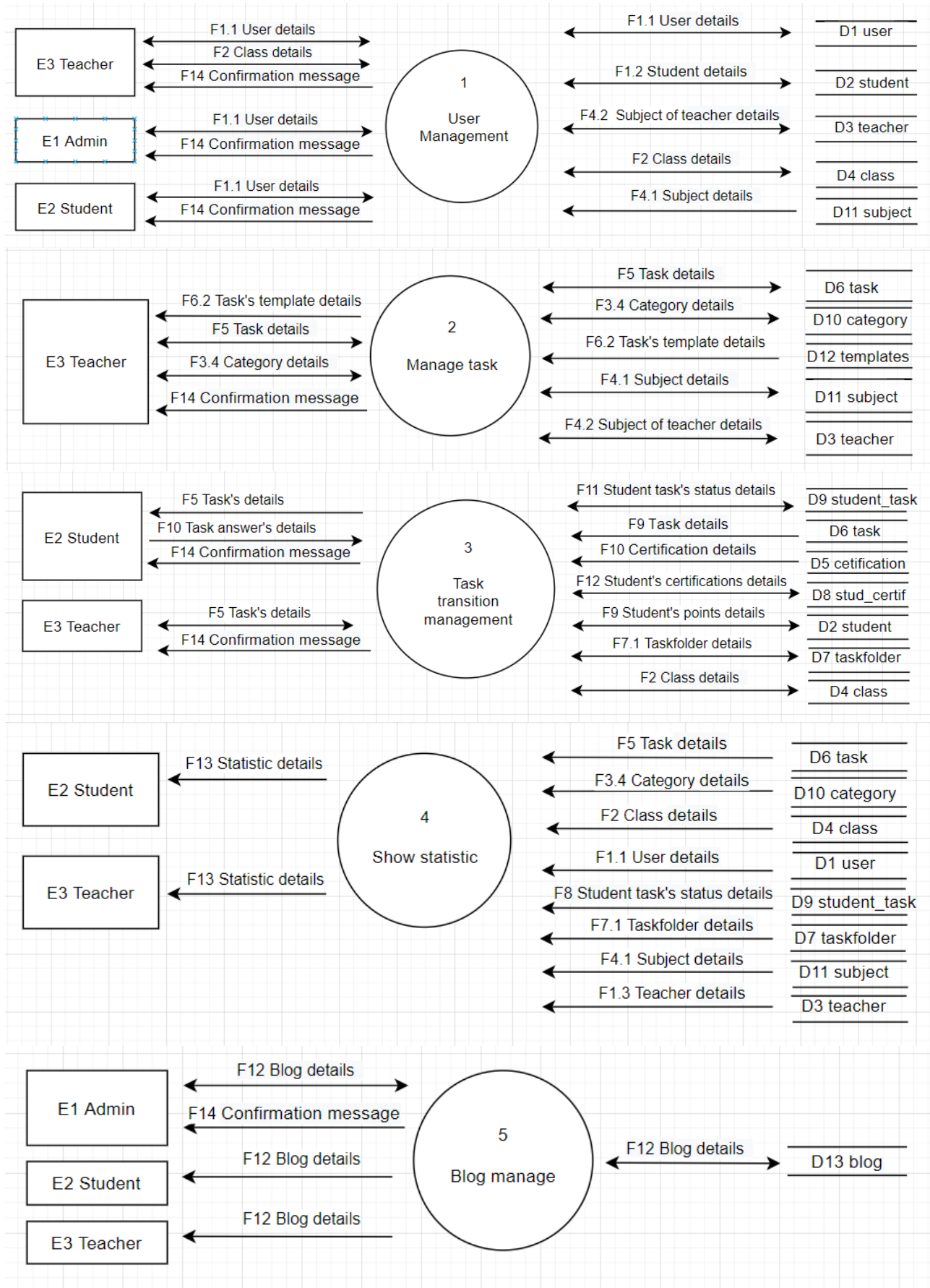
אלגוריתם יותר יעיל

1.2 דיאגרמת הקשר (Context Diagram).

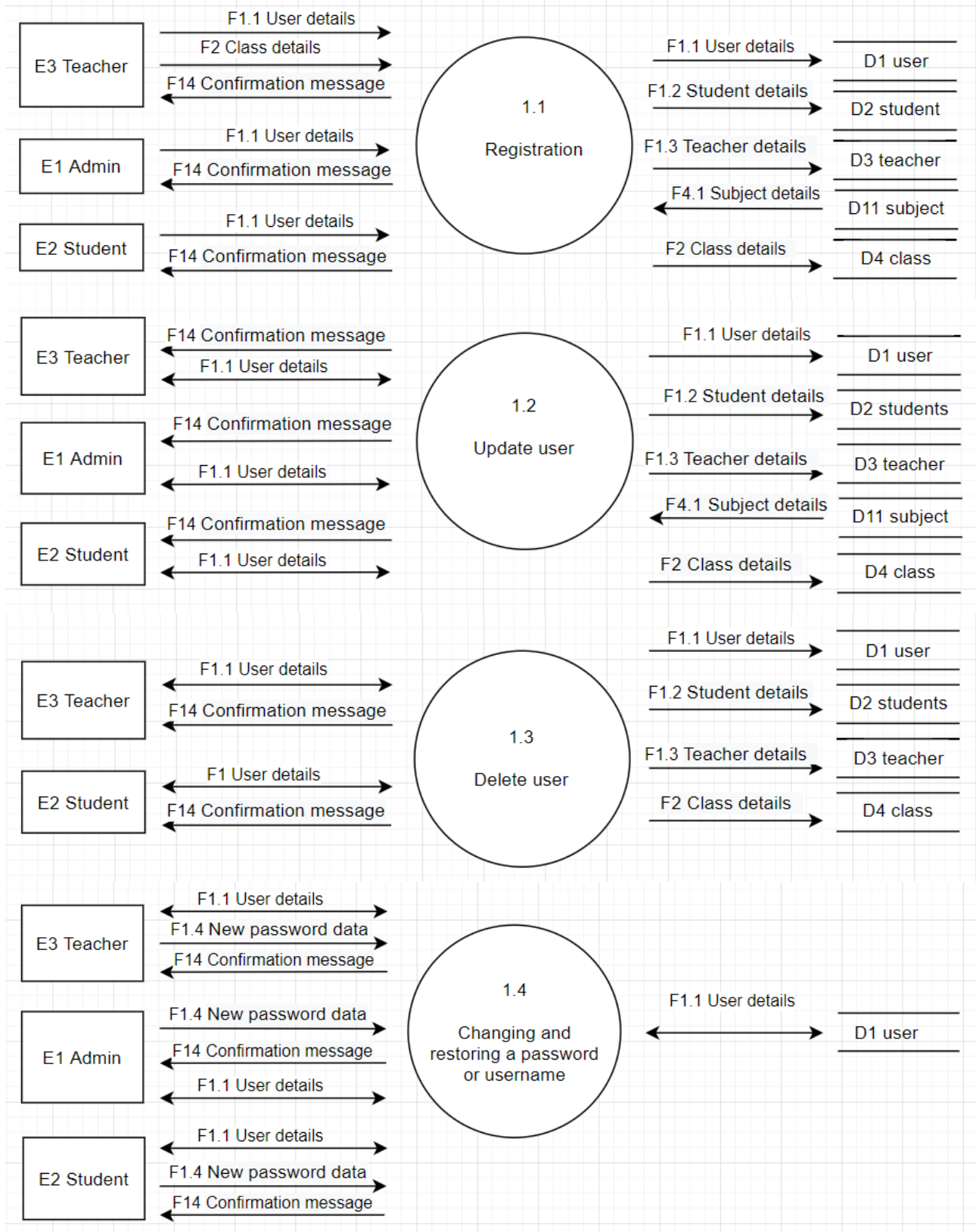


1.3 ניתוח התהליכים העיקריים.

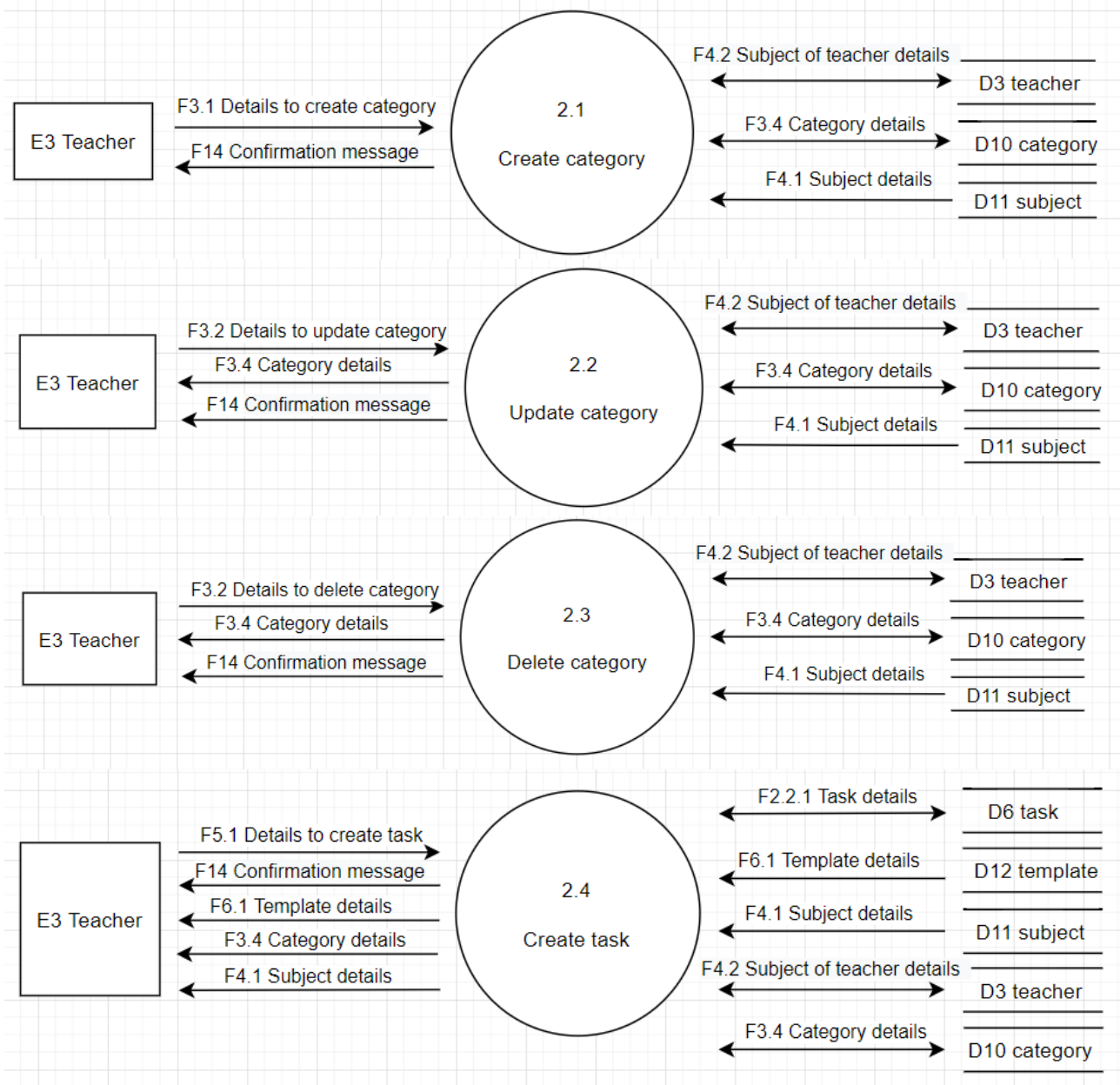
DFD0

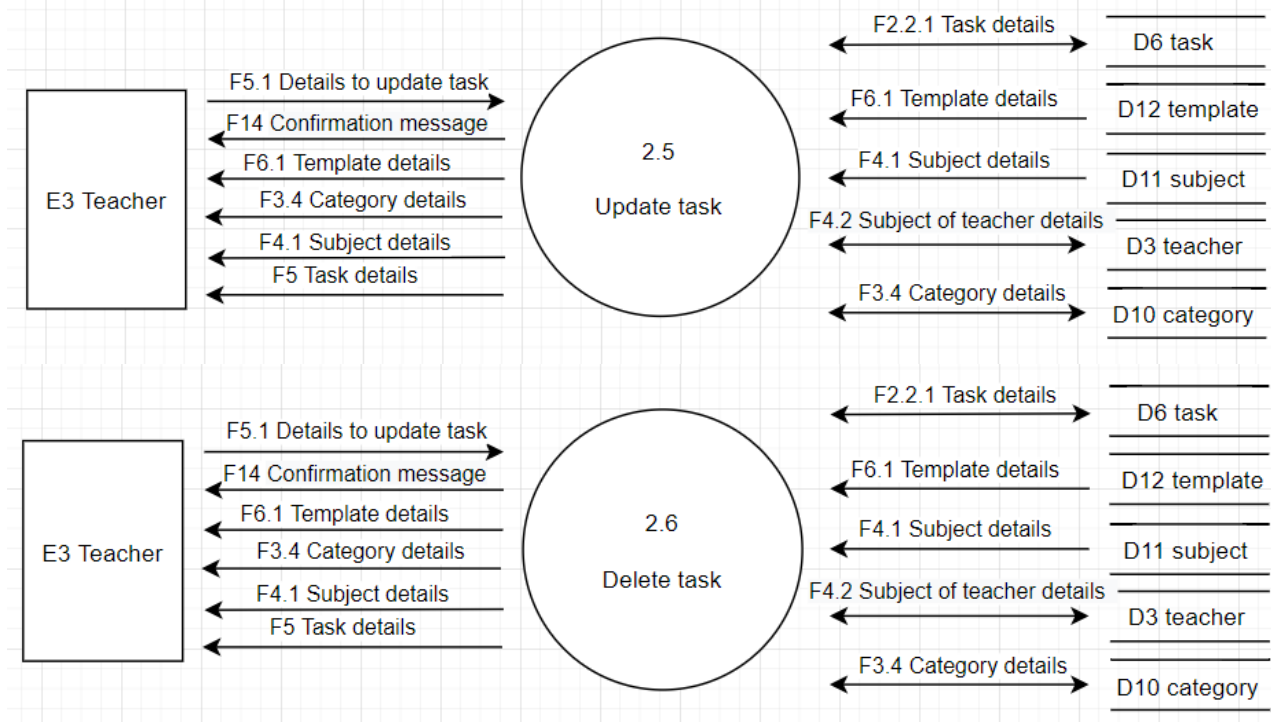


DFD1 – User Management

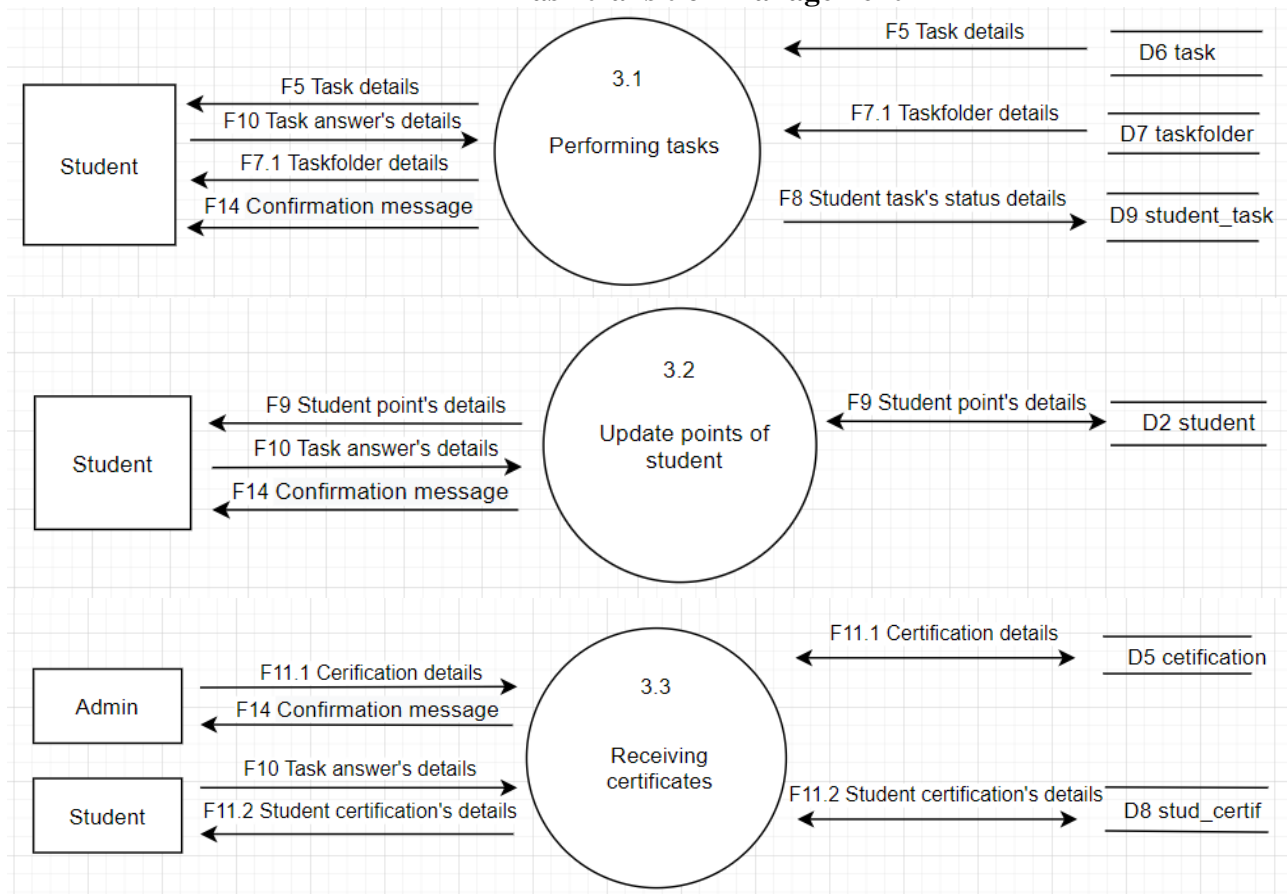


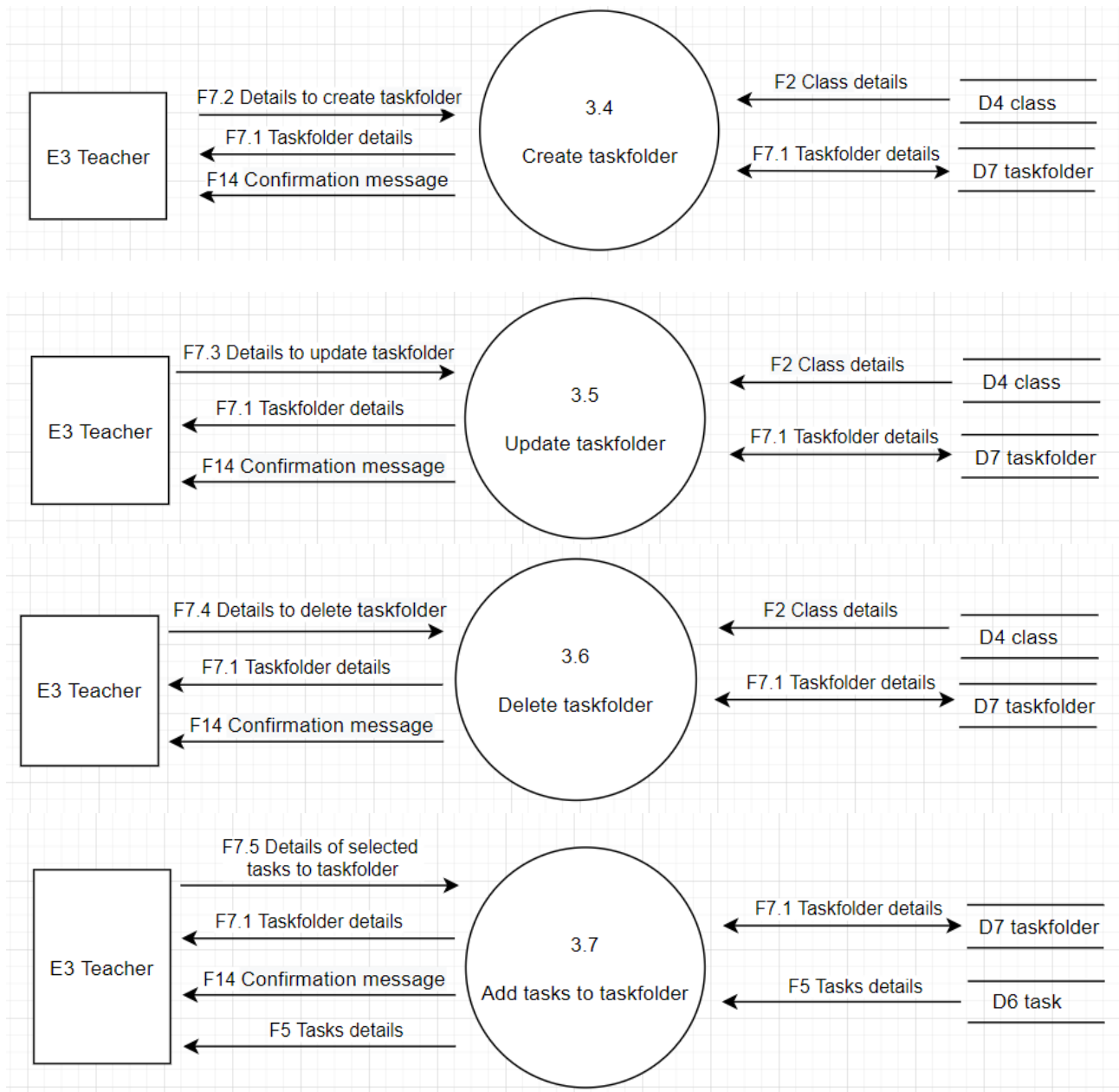
DFD1 – Manage tasks



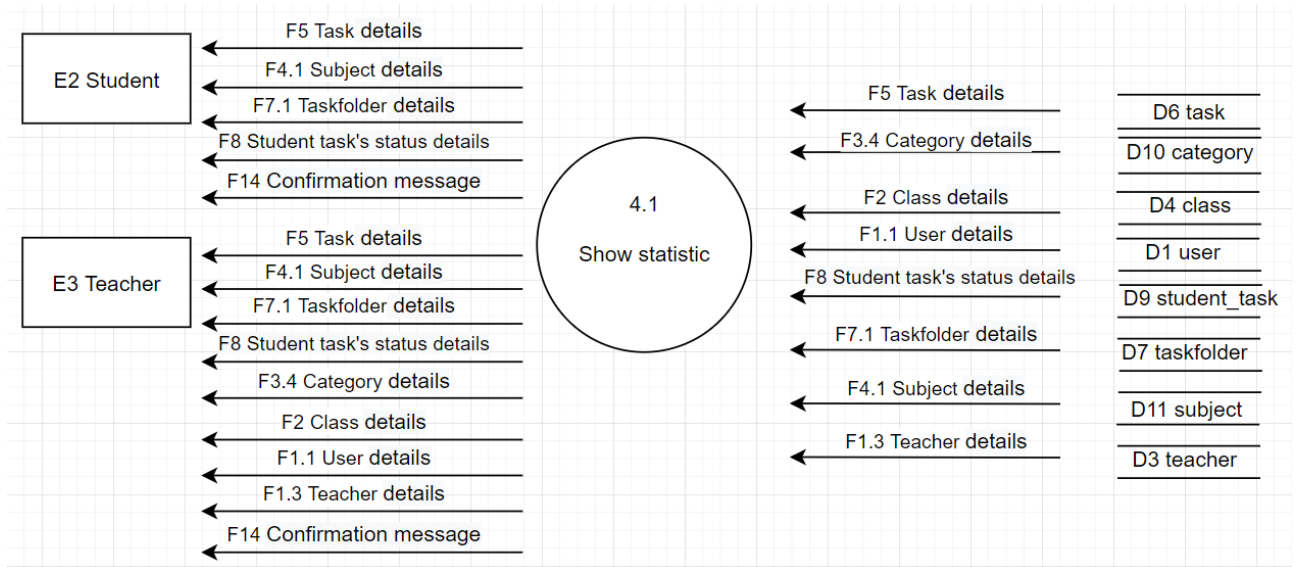


DFD1 – Task transition management

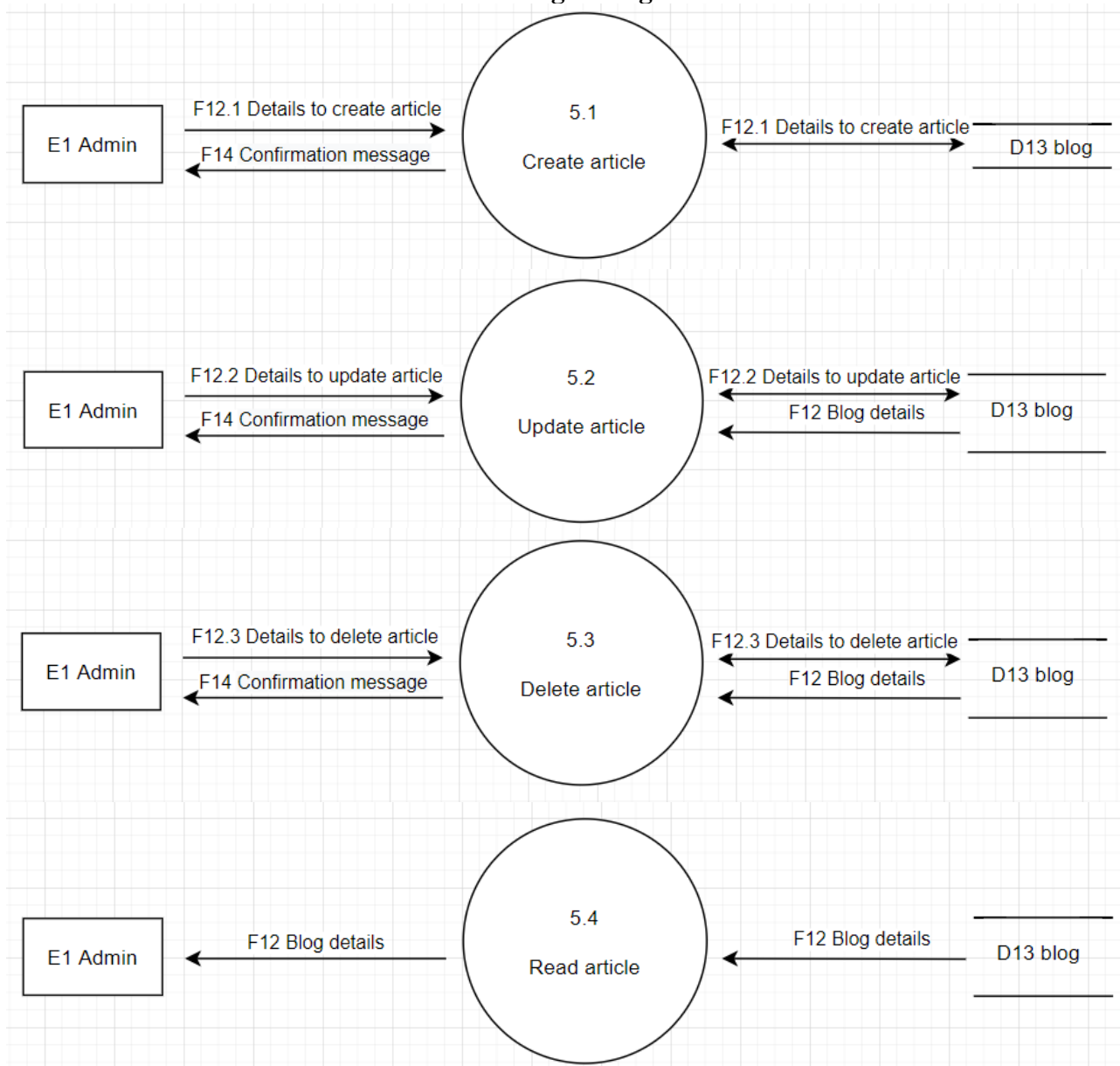




DFD1 – Statitic



DFD1 – Blog management



2 מילון נתונים

2.1 תאור ישויות.

קוד	שם הישות	תיאור
E1	Admin	אחראי על יצירת משימות ורמות
E2	Student	אדם המשתמש במערכת

E3	Teacher	אדם המשתמש במערכת
----	---------	-------------------

2.2 תאור המאגרים.

שדות הטבלה	שם טבלה	קוד + שם מאגר
<u>id_user</u> , role, email, password, name, lastName	user	D1 user
<u>id_user</u> , class_level, total_points	student	D2 student
<u>id_user</u> , <u>id_subject</u>	teacher_subject	D3 teacher
<u>id_class</u> , class_name, class_level, id_teacher	class	D4 class
<u>id_user</u> , <u>id_class</u>	student_class	
<u>id_certif</u> , name_certif, point, img_url	certification	D5 certification
<u>id_task</u> , task_name, task_text, task_create_date, task_weight, task_level, task_img, specific_data, id_teacher , id_category , id_template	task	D6 task
<u>id_task</u> , id_tskFolder	task_taskfolder	
<u>id_tskFolder</u> , tskFolder_name, is_publish, id_class , id_subject	taskfolder	D7 taskfolder
<u>id_user</u> , <u>id_certif</u> , data_get	student_certification	D8 stud_certif
<u>id_user</u> , <u>id_task</u> , is_task_done, date	student_task	D9 student_task
<u>id_category</u> , category_name, date_create, id_subject	category	D10 category
<u>id_subject</u> , subject_name, subject_icon	subject	D11 subject
<u>id_template</u> , template_name, template_img	template	D12 template
<u>id_article</u> , art_title, art_text, art_img, create_date	blog	D13 blog

2.3 תאור זרימות.

שדות הזרימה	יעד	מקור	קוד + שם זרימה
<u>id_user</u> , role, email, password, name, lastName	D1 user	E1 Admin, E2 Student, E3 Teacher	F1.1 User details
<u>id_user</u> , class_level	D2 student	E2 Student	F1.2 Student details
<u>id_user</u> , <u>id_subject</u>	D3 teacher	E3 Teacher	F1.3 Teacher details
<u>id_user</u> , email, password	D1 user	E1 Admin, E2 Student, E3 Teacher	F1.4 New password data
<u>id_class</u> , class_name, class_level, <u>id_teacher</u>	D4 class	E3 Teacher	F2 Class details
<u>id_category</u> , category_name, date_create, <u>id_subject</u>	D10 category	E3 Teacher	F3.1 Details to create

			category
id_category, category_name, date_create, id_subject	D10 category	E3 Teacher	F3.2 Details to update category
id_category	D10 category	E3 Teacher	F3.3 Details to delete category
id_category, category_name, date_create, id_subject	D10 category	E3 Teacher	F3.4 Category details
id_subject, subject_name, subject_icon	E3 Teacher	D11 subject	F4.1 Subject details
id_user, id_subject	D3 teacher	E3 Teacher	F4.2 Subject of teacher details
id_task , task_name, task_text, task_create_date, task_weight, task_level, task_img, specific_data, id_teacher, id_category, id_template	D6 task	E3 Teacher	F5 Task details
id_task , task_name, task_text, task_create_date, task_weight, task_level, task_img, specific_data, id_teacher, id_category, id_template	D6 task	E3 Teacher	F5.1 Details to create task
id_task , task_name, task_text, task_create_date, task_weight, task_level, task_img, specific_data, id_teacher, id_category, id_template	D6 task	E3 Teacher	F5.2 Details to update task
id_task , id_teacher	D6 task	E3 Teacher	F5.3 Details to delete task
id_template, template_name, template_img	E3 Teacher	D12 template	F6.1 Template details
Id_task, id_template	D12 template	E3 Teacher	F6.2 Task's template details
id_tskFolder, tskFolder_name, is_publish, id_class, id_subject	D6 taskfolder	E3 Teacher, E2 Student	F7.1 Taskfolder details
id_tskFolder, tskFolder_name, is_publish, id_class, id_subject	D7 taskfolder	E3 Teacher	F7.2 Details to create taskfolder
id_tskFolder, tskFolder_name, is_publish, id_class, id_subject	D7 taskfolder	E3 Teacher	F7.3 Details to update taskfolder
id_tskFolder, id_class	D7 taskfolder	E3 Teacher	F7.4 Details to delete taskfolder
id_task, id_tskFolder	D7 taskfolder	E3 Teacher	F7.5 Details of selected tasks to taskfolder
id_user, id_task, is_task_done, date	D9	E2 Student	F8 Student task's status

	student_task		details
id_user , total_points	D2 student	E2 Student	F9 Student point's details
id_task , task_name, task_text, task_create_date, task_weight, task_level, task_img, specific_data, id_teacher, id_category, id_template, id_user	P3.2 Update points of student	E2 Student	F10 Task answer's details
id_certif, name_certif, point, img_url	D5 certification	E1 Admin	F11.1 Certification details
id_user, id_certif, data_get	D8 stud_certif	3.3 Reciving certificate	F11.2 Student certification's details
id_article, art_title, art_text, art_img	D13 blog	E1 Admin	F12 Blog details
id_article, art_title, art_text, art_img	D13 blog	E1 Admin	F12.1 Details to create article
id_article, art_title, art_text, art_img	D13 blog	E1 Admin	F12.2 Details to update article
id_article, art_title, art_text, art_img	D13 blog	E1 Admin	F12.3 Details to delete article
id_article, art_title, art_text, art_img	D13 blog	E1 Admin, E2 Student, E3 Teacher	F12.4 Details to read article
id_task , task_name, task_create_date, task_img, id_category, id_user, id_category, category_name, id_subject, id_class, class_name, id_tskFolder, tskFolder_name	E2 Student, E3 Teacher	P4 Show statistic	F13 Statistic details
	E1 Admin, E2 Student, E3 Teacher	Study system	F14 Confirmation message

2.4. תאור תהליכים.

קוד	שם התהליך	תיאור
P1.1	Registration	• מאפשר רישום משתמש
P1.2	Update user	• מאפשר עדכון משתמש
P1.3	Delete user	• מאפשר מחיקת משתמש
P1.4	Changing and restoring a password	• מאפשר לשנות סיסמה • מאפשר שיחזור סיסמה דרך דוא"ל

● מאפשר למורה ליצור קטגוריה	Create category	P2.1
● מאפשר למורה עדכון קטגוריה	Update category	P2.2
● מאפשר למורה מחיקת קטגוריה	Delete category	P2.3
● מאפשר למורה ליצור משימה	Create task	P2.4
● מאפשר למורה עדכון משימה בתנאי שלא נמצאת בתיקייה ציבורית	Update task	P2.5
● מאפשר למורה מחיקת משימה בתנאי שלא נמצאת בתיקייה ציבורית	Delete task	P2.6
● מאפשר לתלמיד ביצוע משימה	Performing tasks	P3.1
● מאפשר עדכון נקודות של התלמיד בגין תשובות נכונות	Update points of student	P3.2
● מאפשר לתלמיד קבלת תעודת הערכה לפי צבירת נקודות	Receiving certificates	P3.3
● מאפשר למורה ליצור תיקיית משימות לכל כיתה בנפרד	Create taskfolder	P3.4
● מאפשר למורה עדכון תיקיית משימות בתנאי שלא העבירה מצב התיקייה לציבורי	Update taskfolder	P3.5
● מאפשר למורה מחיקת תיקיית משימות בתנאי שהמשימה לא התבצעה על ידי אף אחד מהתלמידים	Delete taskfolder	P3.6
● מאפשר למורה להוסיף משימות לתיקיה בתנאי שהיא אינה ציבורית	Add tasks to taskfolder	P3.7
● מאפשר למורה לצפות סטטיסטיקה של ביצוע משימות לפי כיתה	Show statistic	P4.1
● מאפשר למורה לצפות סטטיסטיקה כמה תלמידים ביצעו משימה מסוימת בכל כיתה בניפרד		
● מאפשר לתלמיד לצפות סטטיסטיקת ביצוע משימות לפי מקצועות		
● מאפשר לכל המשתמשים קריאת מאמרים מהבלוג להעשרת ידע	Read articles	P5.1
● מאפשר לאדמין יצירת מאמר	Create article	P5.2
● מאפשר לאדמין עדכון מאמר	Update article	P5.3
● מאפשר לאדמין מחיקת מאמר	Delete article	P5.4

4. חלופות שפת מימוש-



במסגרת ספר הפרויקט חשוב להציג בחינה של מספר חלופות עבור שפה/ות מימוש הפרויקט. הנ"ל צריך לכלול דרישות אותן יגדיר התלמיד בבחירת השפה המתאימה. בין יתר השיקולים ניתן לכלול:

- זמני ריצה

- היבטי אבטחה והגנה,
- הגנה על זכויות יוצרים,
- בינארי או interpreter,

- קלות במימוש

- התאמה לממשק משתמש או צד שרת וכו'

בחרתי לכתוב את הממשק משתמש בעזרת ספריית React ואת צד השרת באמצעות פלטפורמה Node.js.

Node.js הוא זמן ריצה מקורי של Javascript שנבנה על מנוע V8 Javascript של Chrome. הוא משתמש במודל קלט/פלט שאינו חוסם מונע אירועים והוא בנוי להרחבה וביצועים גבוהים.

React.js היא ספריית Javascript חיצונית חנימית, קוד פתוח, הפועלת בדפדפן ומשמשת לבניית ממשקי משתמש. הוא פותח ומתוחזק על ידי פייסבוק.

Node.js היא פלטפורמה נוחה לפיתוח שרתי אינטרנט חזקים עבור יישומי Node.js. React עובד לצד NPM (Node Package Manager), שהוא מאגר מקוון לפרסום פרויקטי Node.js בקוד פתוח שניתן להתקין בקלות באמצעות ממשק שורת פקודה.

Node מאגדת אפליקציית React לקובץ בודד להידור קל באמצעות מודולי צומת כמו Webpack. מנוע ה-V8 ב-Node.js יכול לטפל בבקשות בכמות גדולה באמצעות פונקציות התקשרות חוזרת (V8 הוא מנוע Javascript בתוך Node.js שמנתח ומבצע את ה-Javascript).

Node.js ו-React.js שניהם משתמשים בשפת Javascript שניתן להפעיל הן בצד השרת והן בצד הלקוח ולכן ניתן להפעיל קוד React.js ישירות בסביבת Node.js.

בנוסף, רכיבי React DOM תוכננו במיוחד לעבוד עם Node.js, מה שמפחית שורות קוד ומפשט את העיבוד בצד השרת.

חוץ מזה, שיתוף ספריות אלה מספק את היתרונות הבאים:

1. עוזר לשמור על עומס שרת גבוה

הוספת Node בפרויקט React מקלה בהרבה על שרתים לטפל בבקשות רשת מרובות ולנהל איזון עומסים בשרת.

2. עוזר לשמור על נתונים בזמן אמת

אם האפליקציה דורשת ניהול נתונים בזמן אמת או הזרמת נתונים, השילוב של React ו-Node.js יהיה שימושי ביותר. הסיבה לכך היא ש-Node הוא הטוב ביותר לניהול חיבורים מתמשכים לשרת, בעוד React הוא הטוב ביותר לעיבוד נתונים חיצוניים.

3. עוזר בפיתוח יישומי עמוד בודד

באמצעות React ו-Node.js, אפשר בקלות לבנות יישומי עמוד בודד. ו-React מאפשר ליצור UI/UX אינטואיטיבי, בעוד ש-Node מספק טעינת נתונים אסינכרוני באמצעות התקשרות חוזרת.

שתי הטכנולוגיות המתוארות כאן משלימות זו את זו היטב. על ידי שילובם, אנו מקבלים שילוב רב עוצמה של כלים לפיתוח אתרים אמינים.

5. פירוט בדיקות תוכנה ואופן ביצוען -

כולל את רשימת בדיקות התוכנה, בדיקות יחידה, בדיקות תהליכיות- full Flow במסגרת מסמך תכנון בדיקות ובדיקות.

הנ"ל יוצג בטבלה (השלימו והגדילו אותה לפי הצורך):

תיאור הבדיקה	תוצאה רצויה	תוצאה מתקבלת
בדיקה התחברות למערכת עם פרטים שגויים	הודעת שגיאה למשתמש.	הודעת שגיאה למשתמש שאחד הפרטים לא נכון
בדיקה שחזור סיסמא בעזרת כתובת מייל שלא קיימת במערכת	הודעת שגיאה למשתמש על כך שהמייל שהוזן אינו רשום במערכת	הודעת שגיאה למשתמש על כך שהמייל שהוזן אינו רשום במערכת
בדיקה שאי אפשר להוסיף משתמש אם המייל שהוכנס קיים כבר	המשתמש לא יוכל להירשם עד שיהיה לו מייל ייחודי	המשתמש לא נרשם אלא אם יש לו מייל ייחודי
בדיקה שכל משתמש יוכל לצפות ולעדכן פרטים אישיים בלבד	בעת לחיצה על הכפתור של עדכון נתונים, לסטודנט מופיעים שדות עם נתונים האישיים שלו, שאותם יוכל לעדכן	בעת לחיצה על הכפתור של עדכון נתונים, המשתמש מקבל רק את הנתונים האישיים שלו.
בדיקה שחייב להגדיר בעת הזנת משתמש חדש את הפרטים הבאים: • סוג/תפקיד משתמש (מורה או תלמיד) • שם פרטי • שם משפחה • מייל • סיסמא בנוסף, למשתמש מורה צריך לבחור מקצועות שהוא מלמד ולמשתמש תלמיד – רמת כיתה שהוא לומד כרגע	כאשר משתמש מסיים להאזין את הנתונים ולוחץ על כפתור "הרשמה", על המערכת לוודא שכל הנתונים אכן הוכנסו בצורה נכונה.	המערכת בודקת תקינות הנתונים שהוכנסו, במידה ויש נתונים שחסרים או לא תקינים, המערכת תבקש מהמשתמש להוסיף/לתקן את אותם הנתונים, ורק אז לנסות לשלוח שוב.
בדיקה שלא ניתן להוסיף תלמיד לכיתה בה הוא כבר קיימת, גם בהחנסת דרך קובץ של מיילים של תלמידים	הודעת שגיאה למשתמש על כך שתלמיד כבר קיים בכיתה זו	הודעת שגיאה למשתמש על כך שתלמיד כבר קיים בכיתה זו
בדיקה שאין אפשרות להוסיף תלמיד לכיתה שאינה רשומה במערכת "Funny app", גם בהחנסת דרך קובץ של מיילים של תלמידים	הודעת שגיאה למשתמש על כך שתלמיד לא קיים במערכת	הודעת שגיאה למשתמש על כך שתלמיד לא קיים במערכת
בדיקה שאין אפשרות להוסיף תלמיד לכיתה עם מייל שגוי, גם בהחנסת דרך קובץ של מיילים של	הודעת שגיאה למשתמש על כך שמייל שלו שגוי	הודעת שגיאה למשתמש על כך שמייל שלו שגוי

תלמידים		
בדיקה שאין אפשרות למחוק תיקיה עם משימות עם הסטטוס שלה "ציבורי"	הודעת שגיאה למשתמש על כך שאי-אפשר למחוק תיקיה שכבר הועברה ל"ציבורי"	הודעת שגיאה למשתמש על כך שאי-אפשר למחוק תיקיה שכבר הועברה ל"ציבורי"
בדיקה שאין אפשרות לשנות את התוכן של משימה שכבר נמצאת בתוך תיקיה עם הסטטוס "ציבורי"	הודעת שגיאה למשתמש על כך שאי-אפשר לשנות את התוכן של משימה שכבר נמצאת בתוך תיקיה עם הסטטוס "ציבורי"	הודעת שגיאה למשתמש על כך שאי-אפשר לשנות את התוכן של משימה שכבר נמצאת בתוך תיקיה עם הסטטוס "ציבורי"
בדיקה שתיקיות של משימות עם הסטטוס "לא פורסם" אינן מוצגות בסטיסטיקה	לא מוצגות תיקיות של משימות עם סטטוס "לא פורסם"	לא מוצגות תיקיות של משימות עם סטטוס "לא פורסם"
בדיקה שאין אפשרות למחוק את הקטגוריה אם יש משימה מקטגוריה זו	הודעת שגיאה למשתמש על כך שאי-אפשר למחוק את הקטגוריה אם יש משימה מקטגוריה זו	הודעת שגיאה למשתמש על כך שאי-אפשר למחוק את הקטגוריה אם יש משימה מקטגוריה זו
בדיקה שבחשבון התלמיד, משימות בקטע "שיעורי בית" ייעלמו לאחר השלמתן בהצלחה	לא מוצגות משימות אחרי השלמתן בהצלחה	לא מוצגות משימות אחרי השלמתן בהצלחה

תיאור אלגוריתמים לפתרון הבעיה

אם נעשה שימוש בשילוב של שפות תכנות שונות – יש להסביר כאן
אם נעשה שימוש בתבניות עיצוב (Design Patterns) יש להסביר כאן (כמו MVC, Singleton, Adapter)

כל אלגוריתם לא סטנדרטי או רעיון יעיל מומלץ להביא כאן ולהציג יתרונותיו, כולל חישוב סיבוכיות זמן ו/או מקום.

צריך לתאר 3-4 אלגוריתמים מורכבים. אפשר למקם Activity Diagrams או שאילתות מורכבות (עם תת-שאילתות – לדוגמה) עם הסבר.

המערכת "Funny app" כוללת תבנית **MVC** (Model-View-Controller). להלן הסבר קצר על האופן שבו רכיבי הפרויקט ממפים לתבנית ה-MVC:

מודל: מייצג את הנתונים והלוגיקה העסקית של האפליקציה. בפרויקט, מודלים כוללים טבלאות מוד נתונים (למשל Taskfolder, Student_task) וכל היגיון משויך לאחזור, עדכון ועיבוד נתונים.

תצוגה: התצוגה אחראית להצגת נתונים למשתמש וטיפול באינטראקציה של המשתמש. כלומר, רכיבי ממשק המשתמש שאיתם מנהל המערכת, התלמידים והמורים מקיימים אינטראקציה עם האפליקציה, בוחרים כיתות, מציגים משימות ורואים סטטוס משימות.

בקר: בקרים מנהלים את זרימת הנתונים בין המודל לתצוגה. בפרויקט, רכיבי בקר מנהלים את ההיגיון לקבלת נתונים ממודלים (לדוגמה, קבלת מטלות תלמידים, עדכון סטטוס משימות) והעברת נתונים אלה לתצוגות המתאימות לעיבוד.

Singleton - השתמשנו בתבנית זו למודל שבו אנו מבצעים חיבור לבסיס נתונים. הסיבה לכך

שהשתמשנו ב-singlestone היא כיוון שבעת ביצוע שאילות שונות, אין לנו צורך ביצירת חיבור לבסיס נתונים כל פעם מחדש, בעת קריאה לכל שאילתה. זאת אומרת, שאם ישנו חיבור קיים, אנחנו נשתמש בו, על מנת לבצע שאילתה.

אלגוריתם ב-Node:

```
import mysql from 'mysql';

class DatabaseSingleton {
  constructor() {
    this.connection = mysql.createConnection({
      host: 'localhost',
      user: 'root',
      password: '',
      database: 'test',
    });
  }

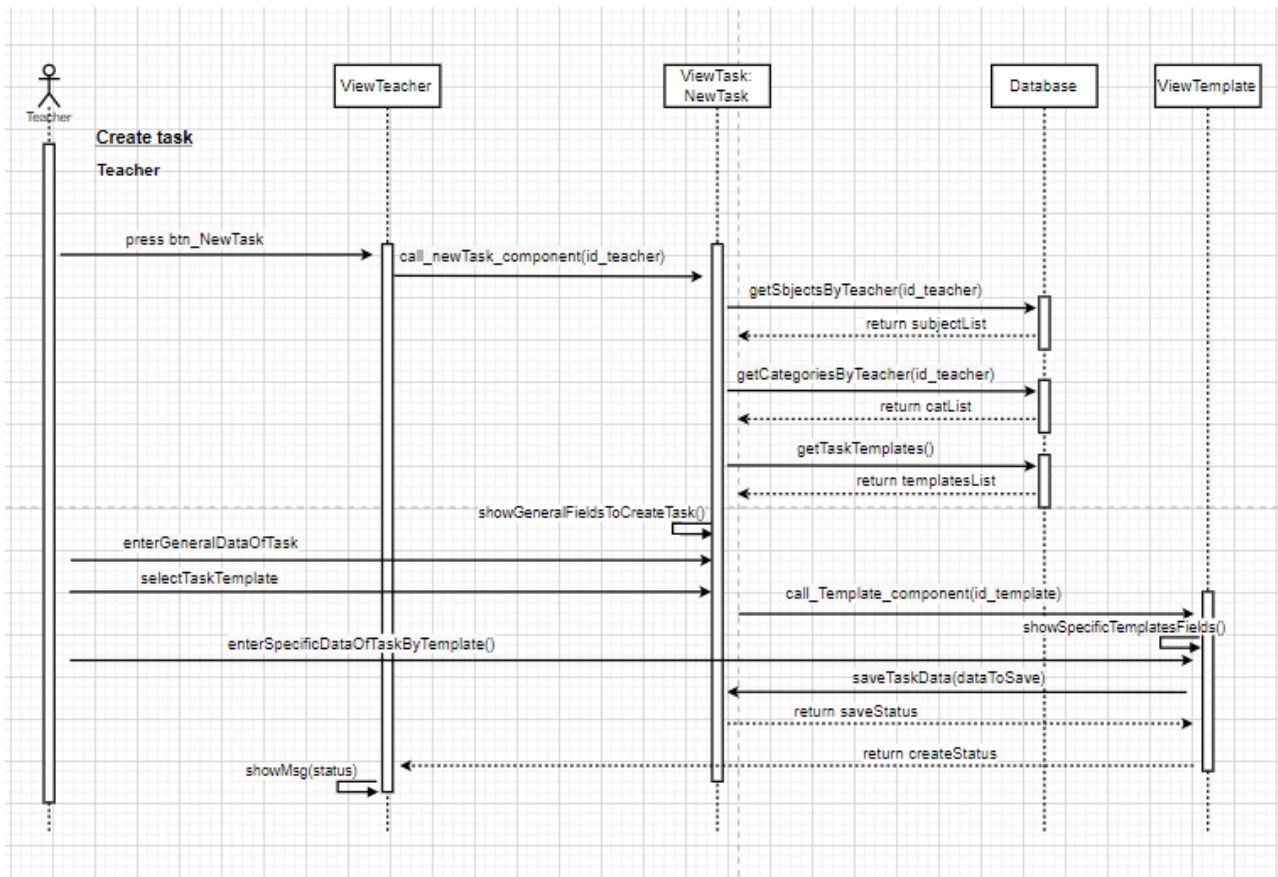
  query(sql, values, callback) {
    return this.connection.query(sql, values, callback);
  }

  static getInstance() {
    if (!this.instance) {
      this.instance = new DatabaseSingleton();
    }
    return this.instance;
  }
}

export const dbSingleton = DatabaseSingleton;
```

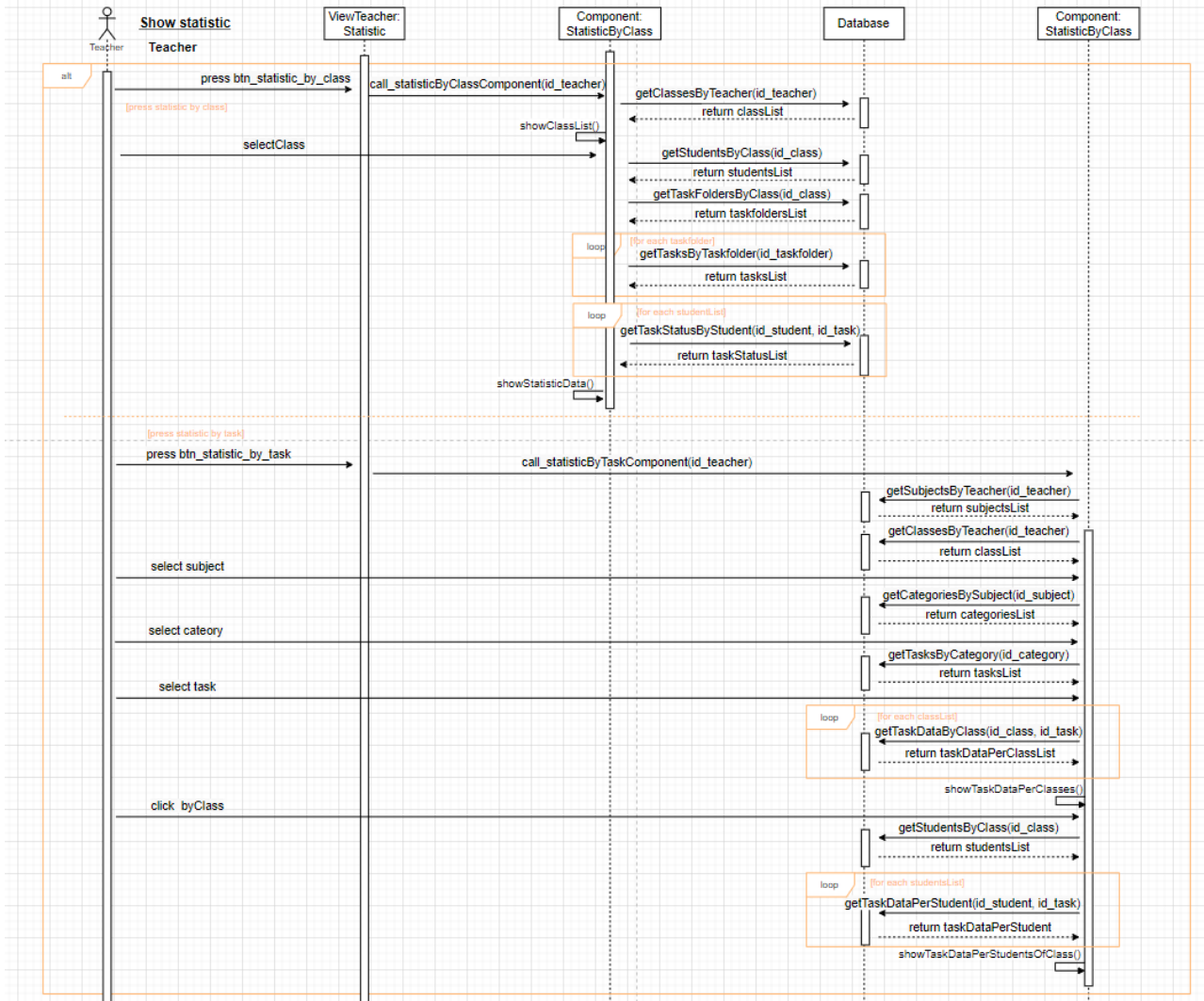
אלגוריתם "יצירת משימה" על ידי המורה

אלגוריתם זה מתאר יצירת משימה עבור תלמיד על ידי מורה. ראוי לציין שתהליך יצירת המשימה עצמה מורכב משני חלקים. 1. מילוי הנתונים המשותפים לכל משימה (נושא, קטגוריה, שם, תיאור, תמונה, רמה, משקל) 2. מילוי נתונים בהתאם לתבנית המשימה שנבחרה. בשלב זה, מרכיב התבנית מחובר, בהתאם לבחירת המורה.



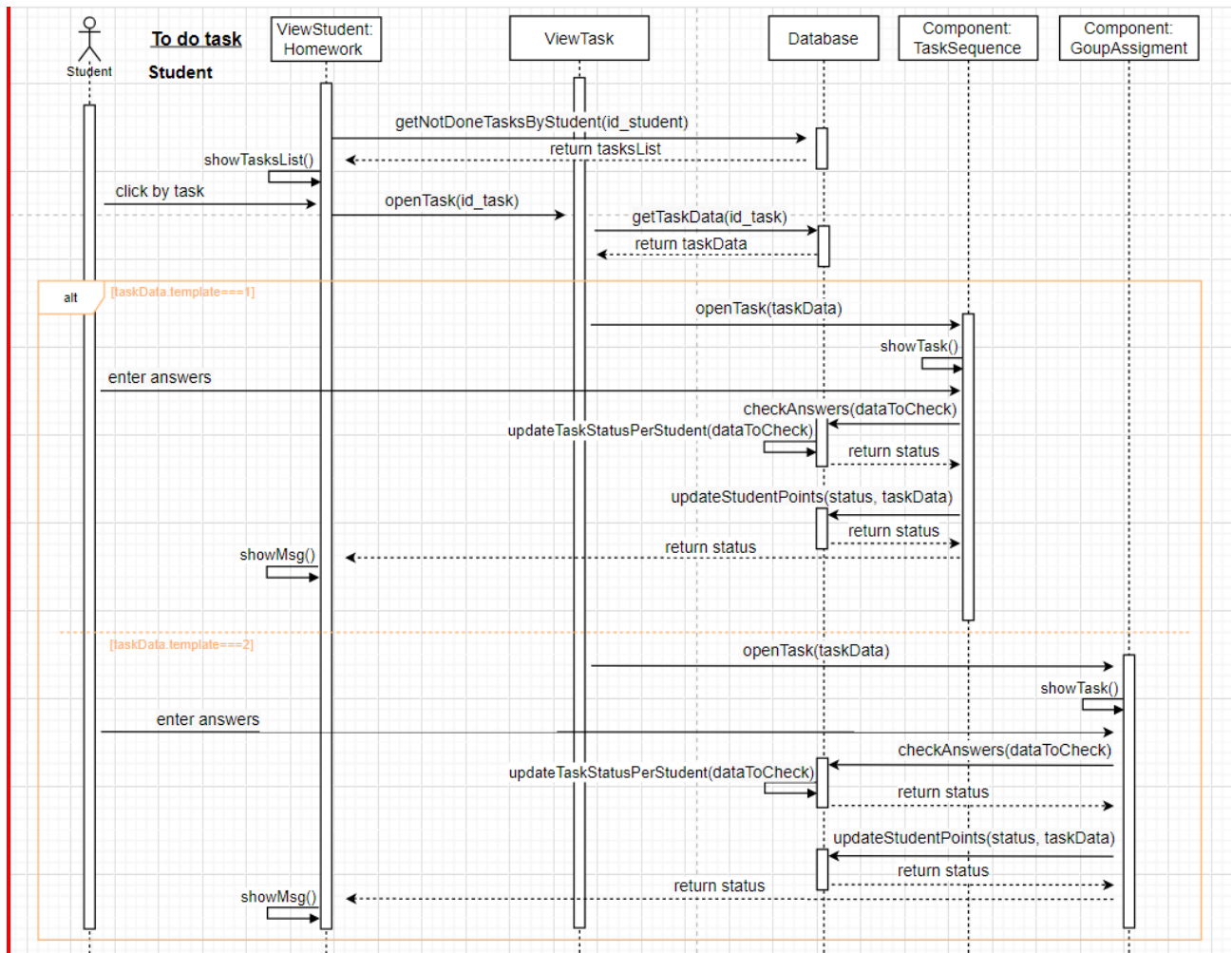
אלגוריתם " לראות סטטיסטיקה" על ידי המורה

האלגוריתם הבא מתאר כיצד המורה משיג נתונים סטטיסטיים. שני סוגים של סטטיסטיקה זמינים עבור המורה: 1. לפי כיתות. כאן המורה מקבל נתונים על כל המשימות שניתנו לכיתה, רואה גם את מצב המשימה לכל תלמיד 2. במשימה. ראשית עליך לבחור את הנושא והקטגוריה של המשימה. כאן המורה רואה כיצד כל כיתה השלימה את המשימה. בנוסף, ניתן לראות סטטיסטיקות מפורטות יותר עבור כל כיתה, כלומר התלמידים שסיימו אותה



אלגוריתם "לבצע משימה" על ידי התלמיד

להלן אלגוריתם לתלמיד לבצע את המשימה שהוטלה עליו על ידי המורה. באלגוריתם זה, בהתאם אשר (TaskSequence, GoupAssignment), מסוימת מתקשרת עם בסיס הנתונים כדי לבדוק את נכונות ביצוע המשימה.



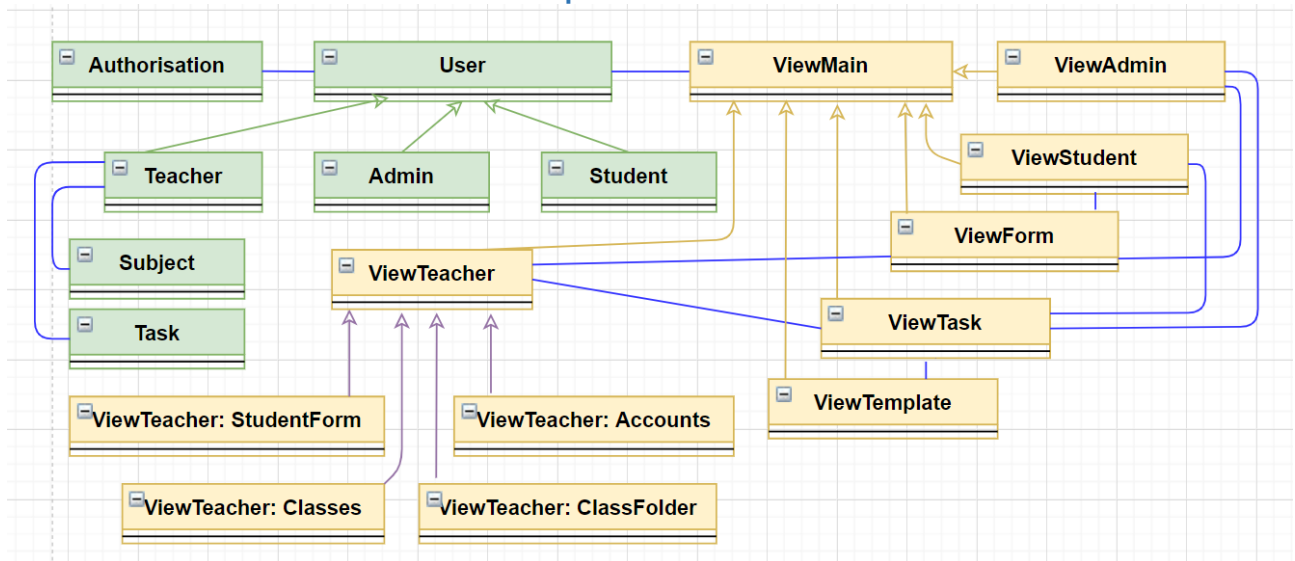
תיאור פתרונות טכנולוגיים

פרק רשות אם יש שימוש בחומרה נוספת, כולל סכמות חיבור והתקנה

מדריך למשתמש בתוכנה

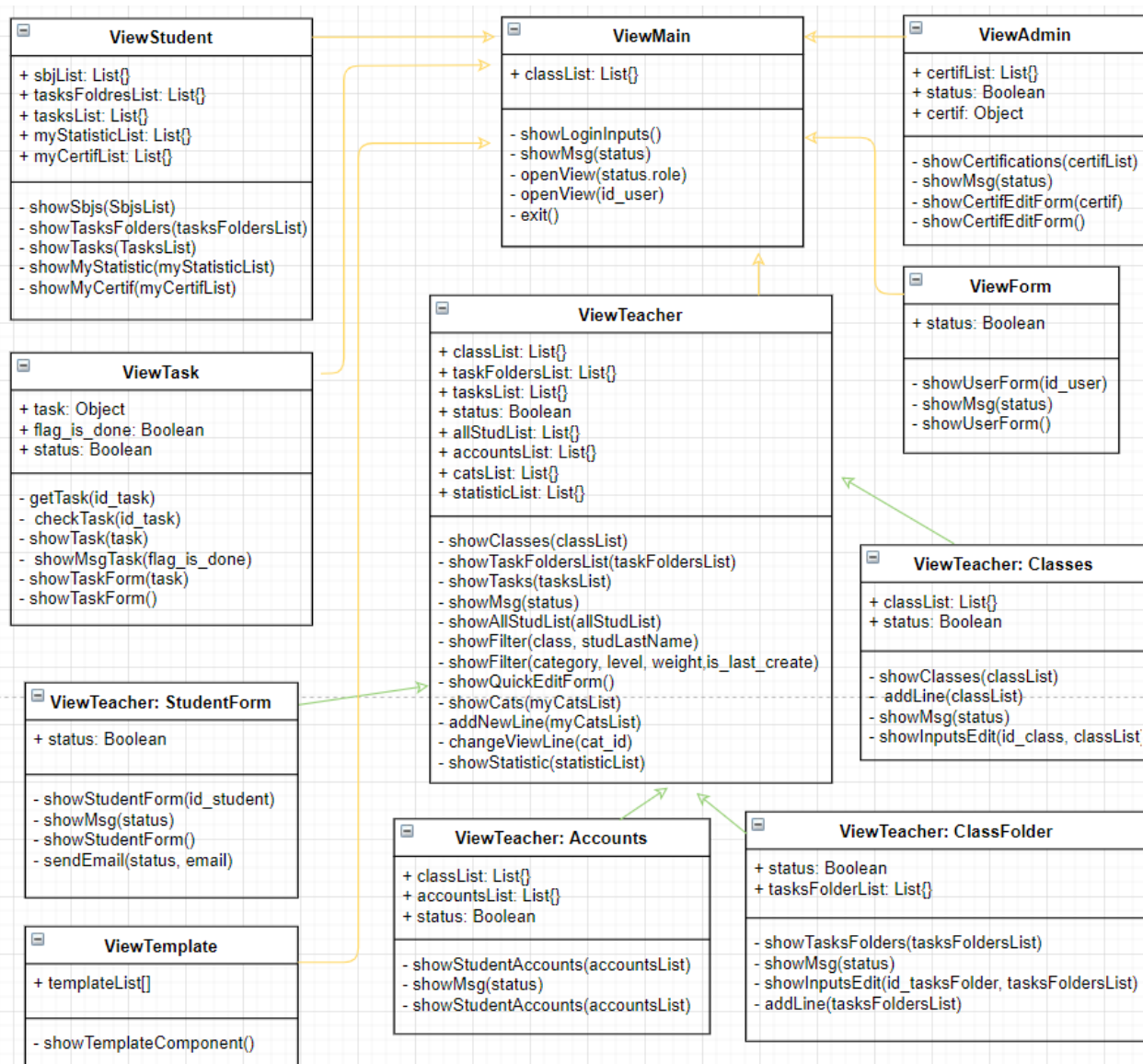
כולל צילומי מסכים והסברים עליהם – המטרה להציג את המערכת שלכם בצורה הטובה ביותר ולחסוך את אלמנט ההפתעה בהגנה על הפרוייקט .

תרשים מחלקות

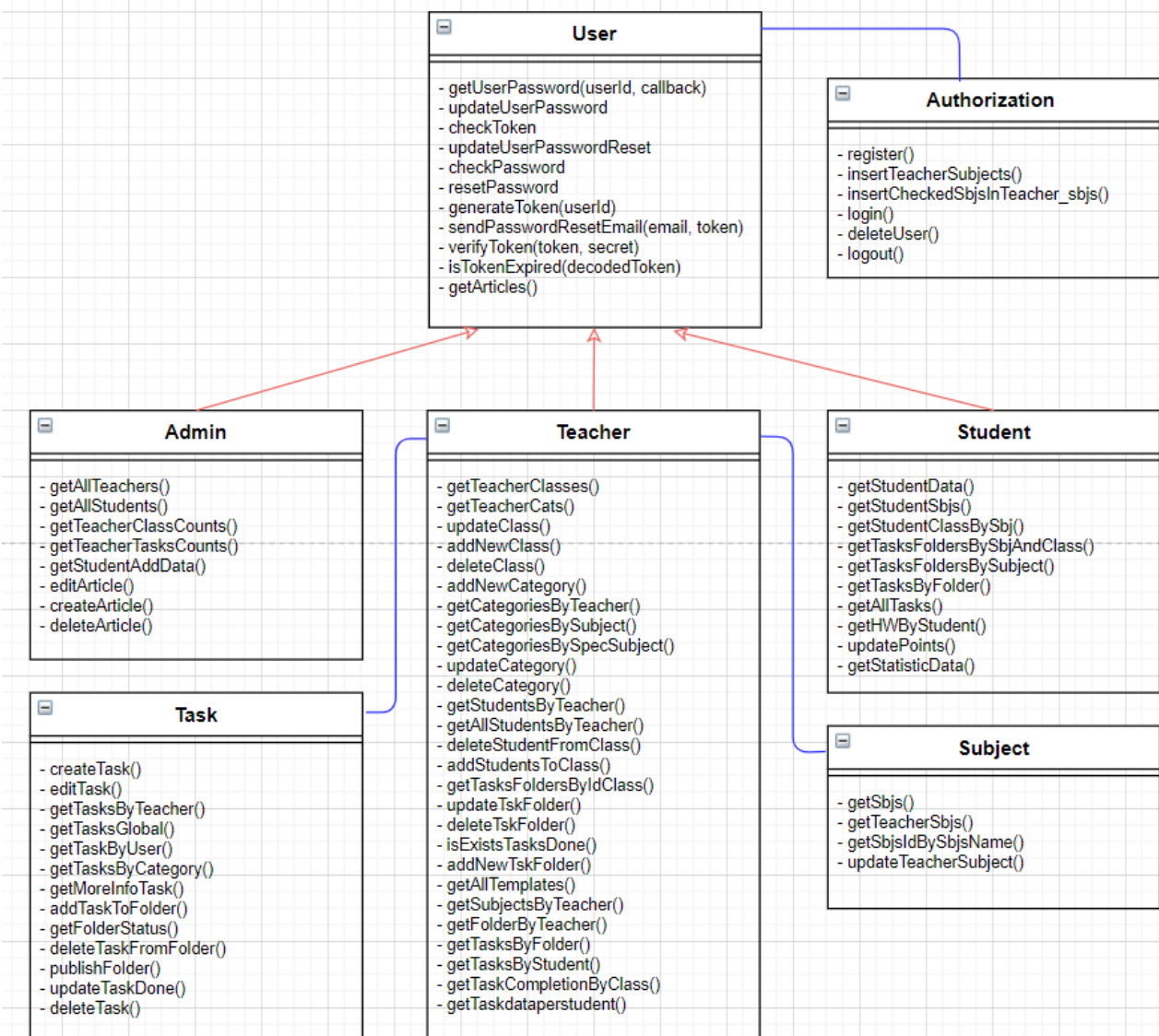


עבור ייצוג חזותי טוב יותר, אנו מחלקים את התרשים לשני חלקים: Backend ו-Frontend

Frontend:



Backend:



התוכניות(כולל דוגמאות הרצה מייצגות).

- בתחילת כל מחלקה חובה לכתוב הסבר קצר על תפקיד של מחלקה
- חשוב להוסיף הערות לכל פונקציה וקטע קוד משמעותי, **אין צורך לתעד כל שורה** .

○ סדר ההדפסה (ברוב הפרוייקטים):

- סדר הצגת מחלקות:
 - מודולים לעבודה עם נתונים BE - (Model)
 - מודולים האחרים על לוגיקה של המערכת BE - (View-Model)
 - מחלקותאו מודולים של ממשק משתמש FE - (View)
- כל מה שאינו מחלקה (פונקציות, אובייקטים נוספים).

אין להוסיף לפרוייקט קוד שאתם לא כתבתם .

שימו לב ! הקפידו על כל שלא תהייה פונקציות ארוכות (מעל 20 שורות קוד) – אם ישנן כאלו, פרקו אותן לפ' משנה .

סיכום ומסקנות

- רפלקצייה אישית - מה למדתם ? האם עמדתם בציפיות של עצמיכם ? (לא חובה)

נספחים

אם יש, פרק רשות (לא חובה)
