**המחלקה להנדסת תוכנה**

**פרויקט גמר – תשע"ח**

**מנוע חיפוש מסעדות לפי מנות**

**Hungry Monkey**

**מאת:**

**שני ג'ופי ודניאל פוטשניקוב**

**מנחה אקדמי: פרופ'/דר'/גב'/מר .......... אישור: תאריך:**

**אחראי תעשייתי: פרופ'/דר'/גב'/מר ........ אישור: תאריך:**

**רכז הפרויקטים: פרופ'/דר' ........ אישור: תאריך:**

**מערכות ניהול הפרויקט:**

|  |  |  |
| --- | --- | --- |
| **#** | **מערכת** | **מיקום** |
| **1** | **מאגר קוד** | **https://github.com/danielpt17/HungryMonkey** |
| **2** | **יומן** | **https://paper.dropbox.com/doc/Project-overview-Hungry-Monkey-vvJDrBjUxaNgpJj3rnYDy** |
| **3** | **ניהול פרויקט (אם בשימוש)** | **Github** |
| **4** | **הפצה** |  |

**-תקציר**

אפליקציית ANDROID לחיפוש מסעדות לפי שם/תיאור של המנה. עם ממשק ניהול וובי לבעלי המסעדות להזנת פרטי המסעדה. האפליקציה תשמש בעיקר את הקהל הרחב לטובת מציאת מה שחפץ לבם בתחום המסעדנות, והן את בעלי המסעדות לטובת פרסום המסעדה.

-מילון מונחים, סימנים וקיצורים

* **Hungry Monkey - שם האפליקציה.**
* **ElasticSearch - מסד נתונים שבו נשתמש.**
* **GUEST - שימוש באפליקציה במצב אורח.**
* **EXIST USER- שימוש באפליקציה במצב משתמש קיים.**

# מבוא

הפרוייקט שלנו הוא מנוע חיפוש העטוף באפלקצית אנדרואיד שמטרתו, לאתר מסעדות לפי סוג/שם המנה.

האפליקציה שלנו באה לתת מענה לאנשים, אשר יודעים בדיוק מה הם רוצים לאכול, אך לא יודעים איפה אותה מנה מוגשת, או לסירוגין, הם ירצו לדעת איפה עוד אפשר לאכול את אותה המנה.

המשתמש יחפש את המנה המסויימת, והאפליקציה תציג לו את כל המסעדות(עם אופציית מיון תוצאות לפי: מרחק /מחיר) המכילות מנה זו.

המוצר יהיה מורכב מצד שרת וצד לקוח:

צד השרת- יכיל מסד נתונים מסוג ELASTICSEARCH- אשר מציע יעילות גבוהה לחיפושים טקסטואליים מורכבים, וימומש בשפת NODEJS.

מסד הנתונים יכיל את תפריטי המסעדות, פרטים על המסעדות ומשתמשי האפליקציה.

כמו כן, יהיה אתר עם ממשק בסיסי למילוי התפריט עבור בעלי המסעדות.

**צד הלקוח(הממשק)**- ימומש בשפת ANDROID, ויתמוך בפיצ'רים הבאים:

* חיפוש בסיסי -איתור מסעדות וברים לפי שם/סוג מנה.
* חיפוש מתקדם- אשר יאפשר חיפוש וסינון לפי פרמטרים נוספים כגון: כשרות, מחיר, מרחק וסוג המסעדה.

את תוצאות החיפוש (מתקדם/בסיסי) ניתן יהיה למיין לפי מרחק/מחיר.

* שימוש באפליקציה במצב **guest או exist user-**

ל- **exist user** נשמור את היסטוריית חיפוש המנות, ואופציה לסמן מנות(אשר מוגשות במסעדה ספציפית) כמנות מועדפות. לעומת **guest**, אשר לו תינתן רק אופציית חיפוש/חיפוש מתקדם.

* **מערכת המלצות- מערכת המלצות ל ,exist userאשר תמליץ על מנות, על סמך אנשים שאהבו מנות דומות.**

# 

# 

# תיאור הבעיה

## -דרישות ואפיון הבעיה

כיום, אחד הבילויים המרכזיים בחיי היום יום שלנו, ומה שמעסיק אותנו במהלך היום הוא: "מה נאכל היום? ואיפה?".

בשוק קיימים מגוון פתרונות לפתירת הבעיה, אשר מציעים לנו שירותי מציאת מסעדות לפי :אווירה, סוג מטבח ומיקום. כמו כן, אם אדם רוצה להזמין אוכל הביתה? הפתרון גם כן קיים.

אך, מה יקרה במצב בו אדם יודע **בדיוק** מה הוא ירצה לאכול/לשתות הערב, וירצה לדעת איזה מסעדות ישנן באזור אשר עונות לצרכיו, ולפי זה לבחור את מקום הבילוי? או לסירוגין הכרת מקומות חדשים אשר גם שם הוא יוכל להנות ממנה דומה.

## **-**הבעיה מבחינת הנדסת תוכנה

* עבודה בצוות- סנכרון ותיאום .
* תכנון המסד נתונים בצורה יעילה ביותר, כך שיוכל להציג תוצאות עבור חיפוש בסיסי, ומתקדם (מנה ספציפית או סוג מאכל).
* מקסום תוצאות חיפוש מתאימות. כלומר: הצגת מנות זהות עם שמות שונים. לדוגמה: שקשוקה ירוקה וצרפתית, הן אותן מנות עם שמות שונים.
* הזדהות ללוקחות שמורים מול צד שרת(sessions).
* עבודה בטכנולגיות/שפות חדשות. במקרה שלנו: מימוש מסד נתונים מסוג ES, וצד שרת בNODEJS ואנדרואיד.
* מערכת המלצות, אשר תלמד את תוצאות החיפוש של המשתמש הקיים ותציג לו הצעות אשר יכולות להתאים לו.
* מציאת שרת אחסון לאפליקציה.

# תיאור הפתרון

פרק זה יכיל תיאור הפתרון המוצע: תרשימי שימוש, ארכיטקטורת המערכת, אלגוריתמים , תיאור הקוד ותיכון (תרשימי הפצה ומודולים עיקריים, שפות וכלים מתוכננים לשימוש כגון מערכות הפעלה, שימוש ברכיבים קיימים, סביבות, כלי בדיקה).

פרק זה יכיל את המרכיבים העיקריים לתיאור הפרויקט. לפירוט נוסף יש להפנות לנספחים מטה (ראו שם).

## -מהי המערכת

**המערכת שלנו מורכבת משני חלקים: אפליקציית אנדרואיד, וממשק ניהול וובי.**

**הפיצ'רים של האפליקציה הינם:**

* הזדהות משתמש קיים מול צד שרת או שימוש במצב "אורח" .
* אופציה לחיפוש בסיסי - כל המסעדות אשר מגישות את המנה עם אותו השם.
* אופציה לחיפוש מתקדם - עם מאפייני חיפוש וסינון כגון : כשרות ,מחיר, מרחק וסוג מסעדה.
* הצגת היסטורית חיפוש מנות, ומקומות מועדפים למשתמש הקיים.
* מיון תוצאות חיפוש לפי מחיר ומרחק.
* מערכת המלצות למשתמש הקיים, אשר תתבסס על מקומות מועדפים של משתמשים אחרים עם העדפות דומות, ותדע להמליץ למשתמש על מנות חדשות שיכולות לעניין אותו.
* טופס הזנת התפריט עבור בעלי המסעדות עם ממשק WEB.
* נגישות- הגדלת הפונטים, חיפוש קולי.
* הוראות הגעה- קישור לgps.

**הפיצ'רים של ממשק הניהול הינם:**

* הזנת ועדכון פרטי המסעדה והתפריט.
* הפקת קובץ תפריט עבור הפרטים שהוזנו.
* העלאת תפריט קיים.

## -תהליכים ונתוני המערכת

* חיפוש בסיסי- האפליקציה תציג את כל המסעדות באזור, אשר קולעות לחיפוש. את התוצאות ניתן יהיה למיין לפי מרחק ומחיר.
* חיפוש מתקדם - האפליקציה תציג את כל המסעדות באזור על פי פרמטרים כגון: סוג מסעדה, כשרות, טווח מחירים ומרחק. את תוצאות החיפוש גם כן יהיה ניתן למיין לפי מחיר/מרחק. לדוגמה: מהזול ליקר או מהקרוב לרחוק.

## -תיאור הפתרון המוצע

הפתרון יהיה אפליקציית אנדרואיד, שתיתן למשתמש את האופציה לחפש מסעדות, אשר מגישות את המנה הרצויה, ע"פ שמה או סוגה, או ע"פ פרמטרים נוספים כגון: טווח מחירים ,מרחק, סוג מסעדה. האפליקציה תציג לו את המסעדות העונות על פרמטרים אלו .

תהיה למשתמש אפשרות למיין את התוצאות שהתקבלו לפי : מרחק /מחיר.

כמו כן, תהיה מערכת המלצות למשתמש הקיים אשר תמליץ לו על מנות דומות שהוא עשוי לאהוב.

עבור הזנת פרטי המסעדות והתפריטים יהיה ממשק ניהול לבעלי המסעדות, ובעצם ניצור אינטרס פרסום עבורם, וכך בעצם הפליקציה תשרת את שני הצדדים.

## -תיאור הכלים המשמשים לפתרון

הכלים שנשתמש בהם בפרויקט הם: android studio כסביבת עבודה לכתיבת צד לקוח(אפליקציית ממשק למשתמש) בשפת JAVA.

לכתיבת צד שרת נשתמש JETBRAINS כסביבת עבודה לצד שרת בשפת NODEJS. נשתמש גם כן במסד נתונים ELASTICSEARCH ו kibana כממשק ניהול עבור המסד נתונים(ES), redis כמסד נתונים עבור מערכת ההמלצות.

עבור ניהול קוד - אנו נעזרים בGITHUB.

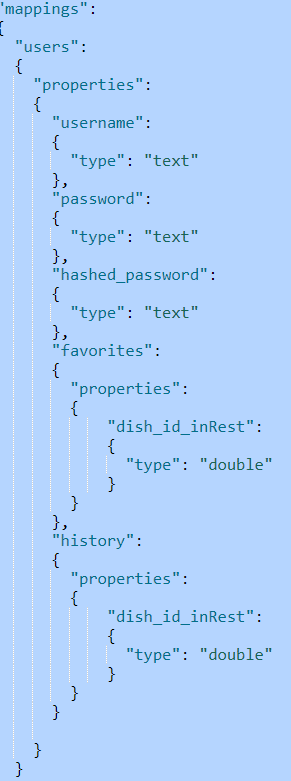
עבור ניהול יומן - dropbox paper.

**-ארכיטקטורת המערכת ומודלים עיקריים**

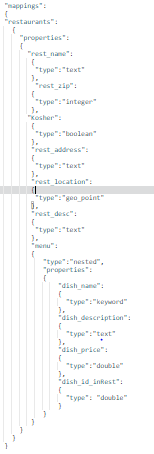
ארכיטקטורת מסד הנתונים בELASTICSEARCH:

מורכב משני חלקים:

עבור השתמשים:



עבור המסעדות:

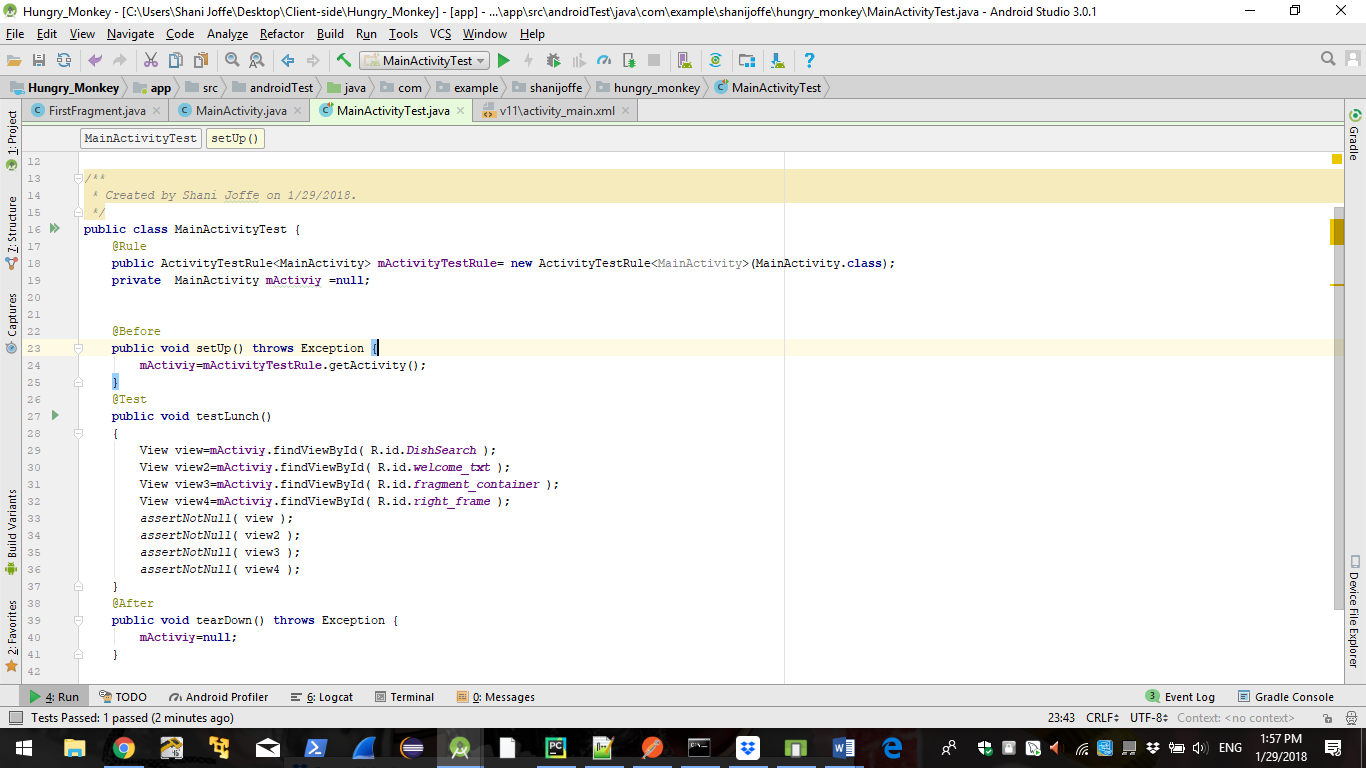


# בדיקות:

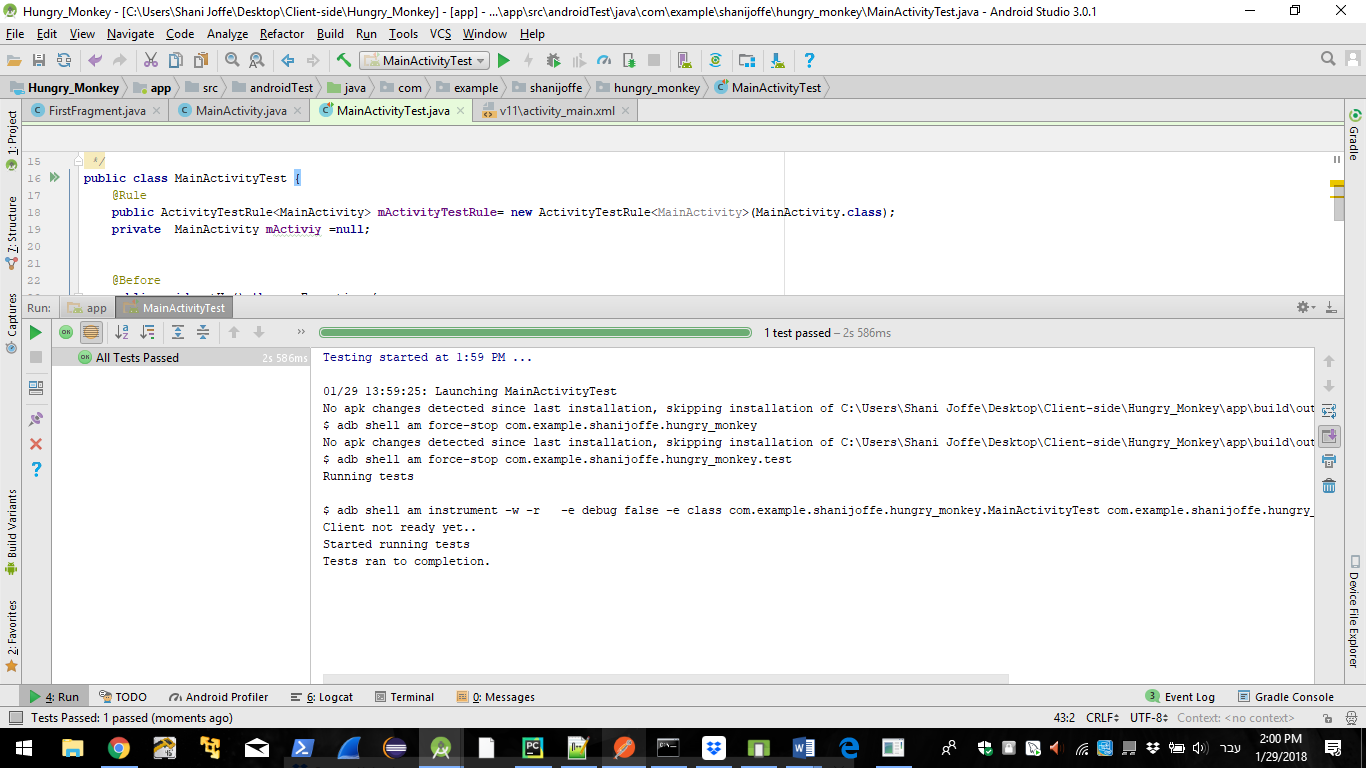
בפרוייקט זה נבדוק כמה סוגי בדיקות :

**Test unit**- נבדוק ACTIVITY מסויים, נכתוב לו TEST ונריץ את הבדיקה. נראה האם ה ACTIVITYעובד באופן תקין כאשר הוא עצמאי.

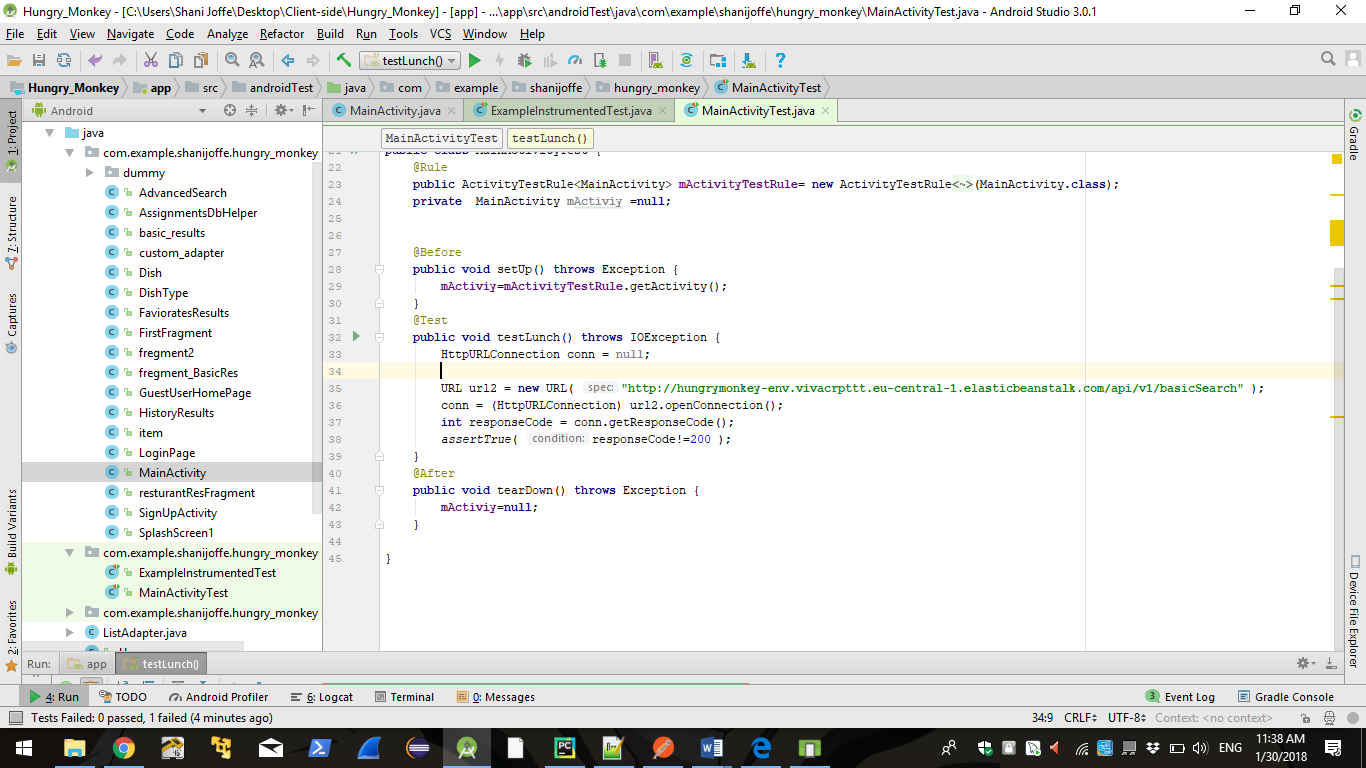
לדוגמה:



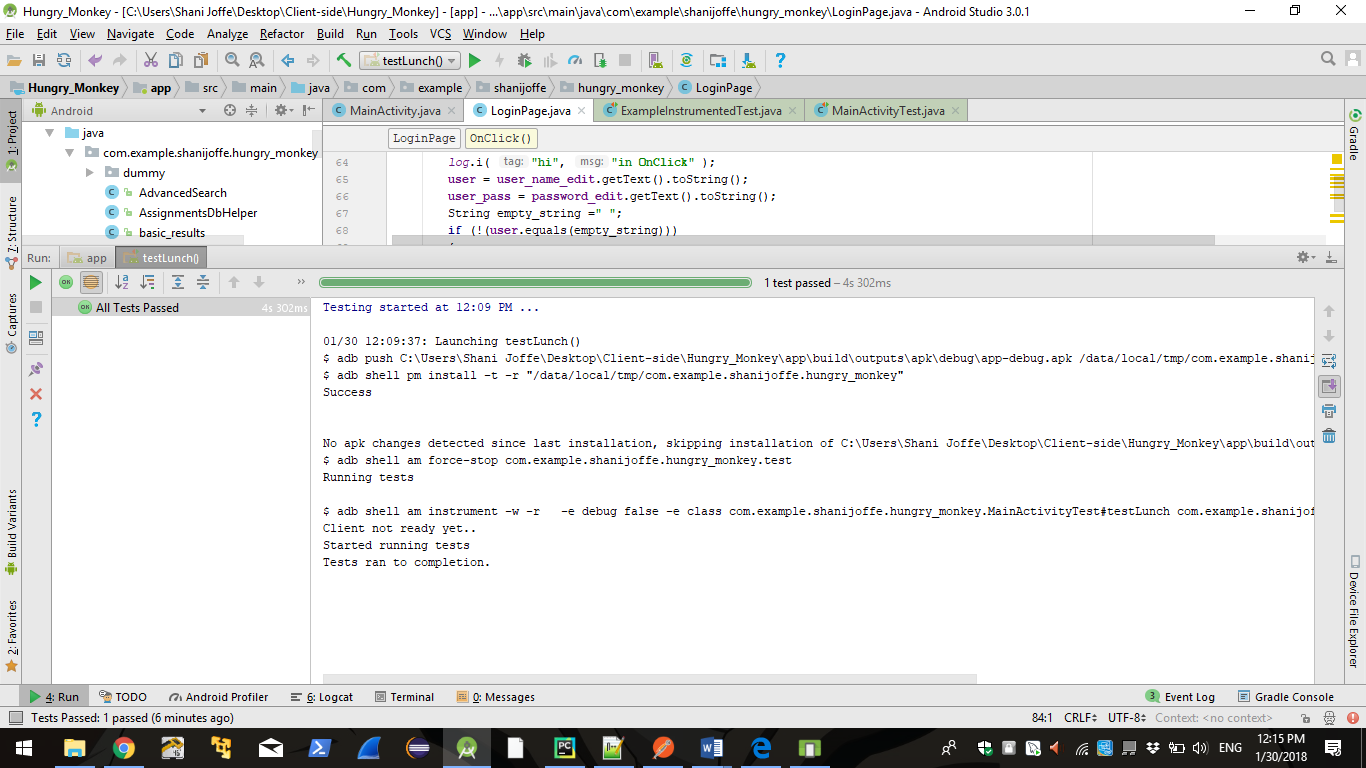
בדוגמה כאן אנו בודקים תקינות רכיבים בMainActivity.xml.

פלט: 

בדיקת test unit- לבדיקת חיבור ל API של החיפוש הבסיסי תקין.

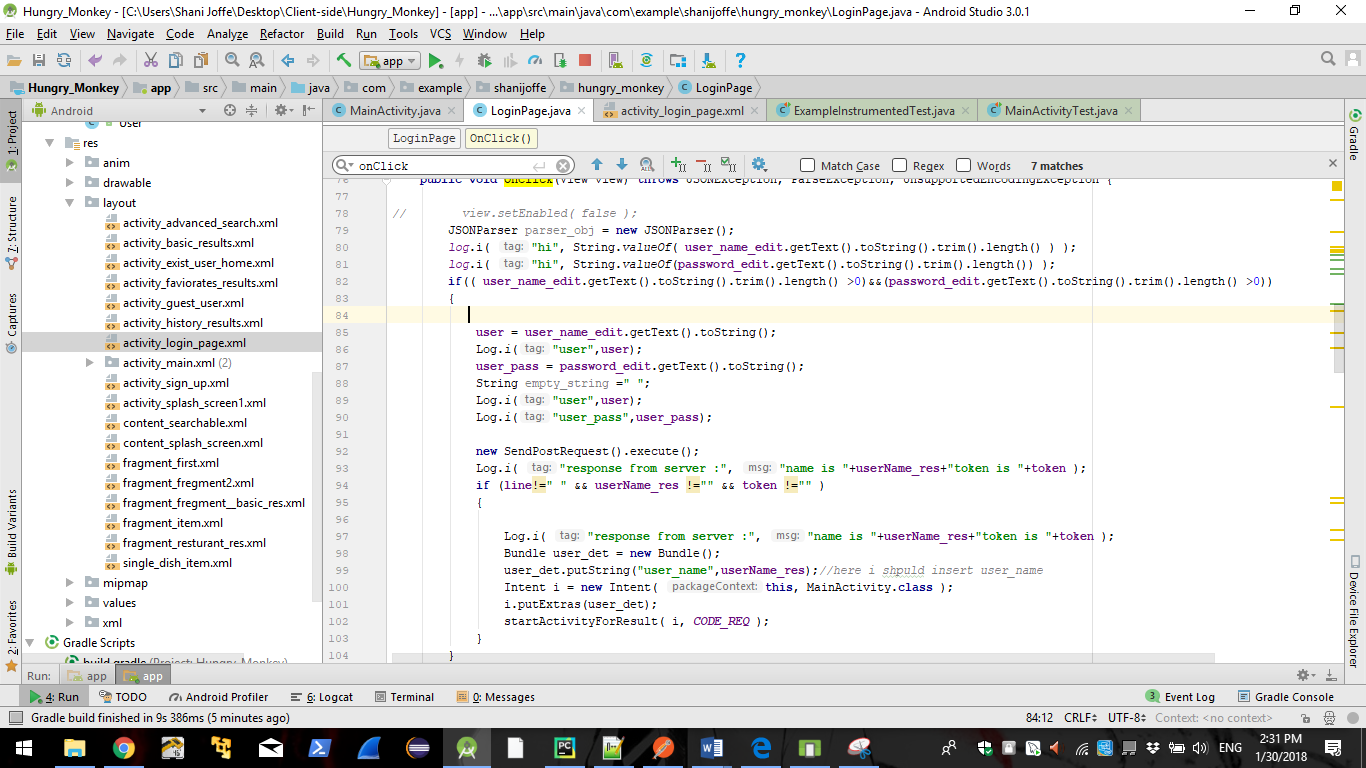


פלט:



**בדיקת אינטגרציה**- בעיקר באפליקציית Android כאשר המידע שהוכנס בActivity אחד צריך להגיע ולהיבדק גם ב Activitiesאחרים. בדיקת האינטגרציה תבדוק שכל המידע עובד ביחד בצורה תקינה. בדיקה זו תעשה לאחר שאסיים את כל הבדיקות יחידה.

בתוך הLoginPage.activity -(דף התחברות באפליקציה ) : נבדוק שהמשתמש אכן הכניס שם משתמש וסיסמא .



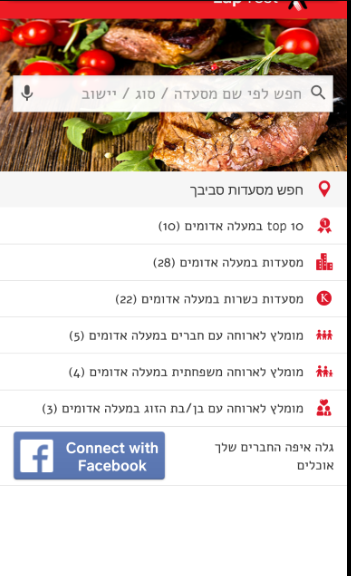
.

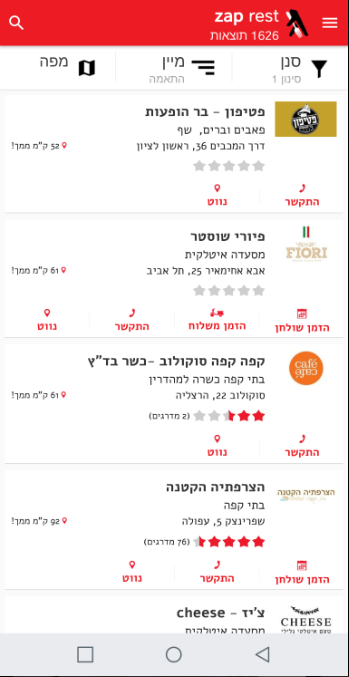
# סקירת עבודות דומות בספרות והשוואה

**-אפליקציות שדומות לאפליקציה שלנו :**

**זאפ רסט(zap rest)**- מנוע חיפוש מסעדות לפי: שם, סוג אוכל ומיקום .

האפליקציה מצגיה למשתמש את המסעדות הקרובות לאזוריו .

****

****

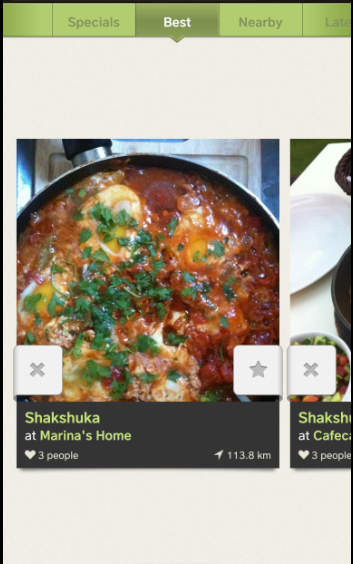
תוצאות חיפוש, עם אופציית מיון וסינון תוצאות.

אפליקציית **FOODSPOTTING -** אפליקציה אמריקאית לחיפוש מסעדות לפי מנות, הדומה ביותר לשלנו, עם תמיכה בחיפושים בשפה העברית ואנגלית, ועם אופציות סינון לפי: מרחק, מיוחדים וכד' .

תוצאות החיפושים בארץ עבור חיפושים בשפה העברית והאנגלית מאוד דלים(למשל עבור חיפוש "שקשוקה" הציגו רק 3 חיפושים בכל הארץ, באנגלית קצת יותר).

כמו כן, אנו נתמוך בחיפוש מתקדם אשר יתאים יותר לקהל הישראלי(כגון : אופציות סינון לפי כשרות).

ונוסיף מערכת המלצות.

****

תוצאות חיפוש באפליקציה עבור שקשוקה .

# נספחים

## -רשימת ספרות \ ביבליוגרפיה:

NodeJs-

<https://books.google.co.il/books?id=8kTCAwAAQBAJ&printsec=frontcover&dq=nodejs&hl=iw&sa=X&ved=0ahUKEwj6qLCfq7bXAhWNDewKHe21ASkQ6AEILzAB#v=onepage&q=nodejs&f=false>.

Elasticsearch DB-

<https://books.google.co.il/books?id=v08oDwAAQBAJ&printsec=frontcover&dq=elasticsearch&hl=iw&sa=X&ved=0ahUKEwiatLDBq7bXAhXE16QKHUyxAC4Q6AEINTAC#v=onepage&q=elasticsearch&f=false>.

AndroidStudio

<https://books.google.co.il/books?id=9ukkDQAAQBAJ&printsec=frontcover&dq=android+studio&hl=iw&sa=X&ved=0ahUKEwjftvbjq7bXAhXPzaQKHT8qBYYQ6AEIOzAC#v=onepage&q=android%20studio&f=false>.

## 

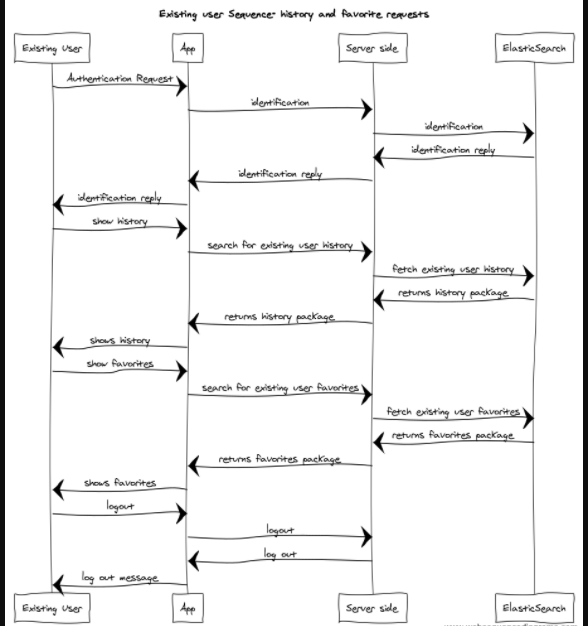
## 

## 

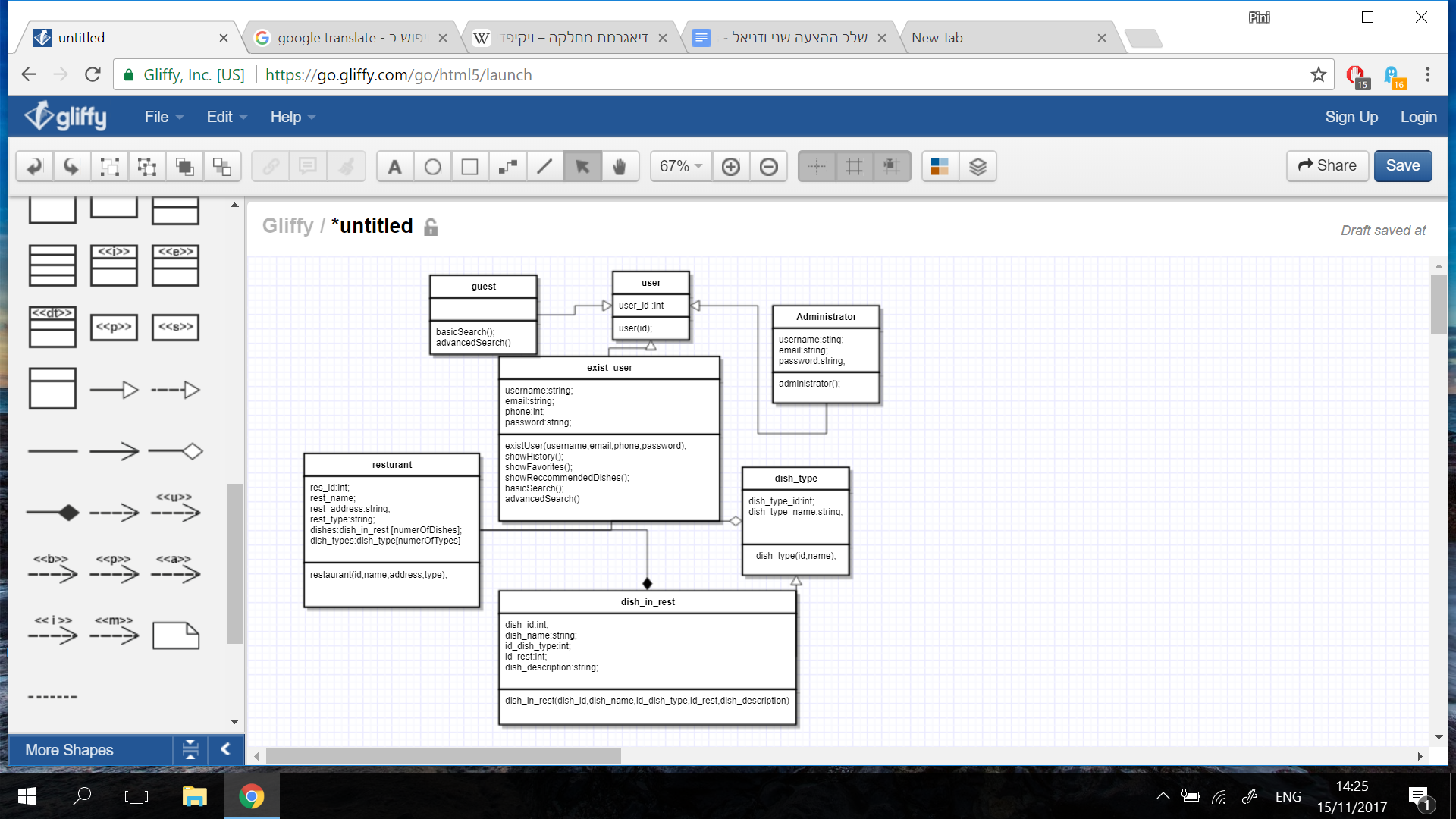
## 

## -תרשימים וטבלאות

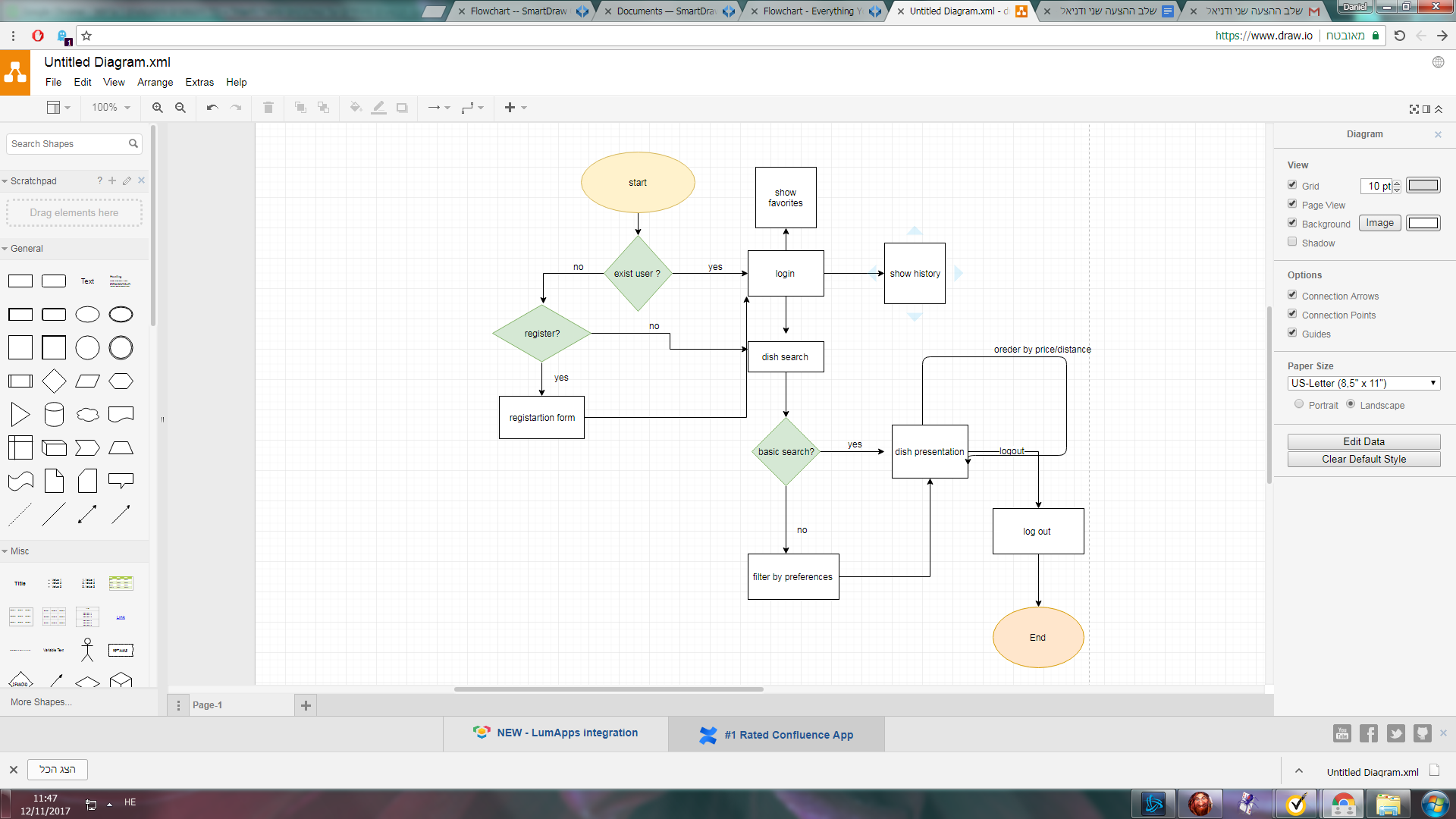
תרשים מצבים - מצב בו המשתמש הקיים מתחבר למערכת, ומבקש לראות את חיפושיו האחרונים וצפייה במסעדות אשר סומנו כמעודפות עליו .



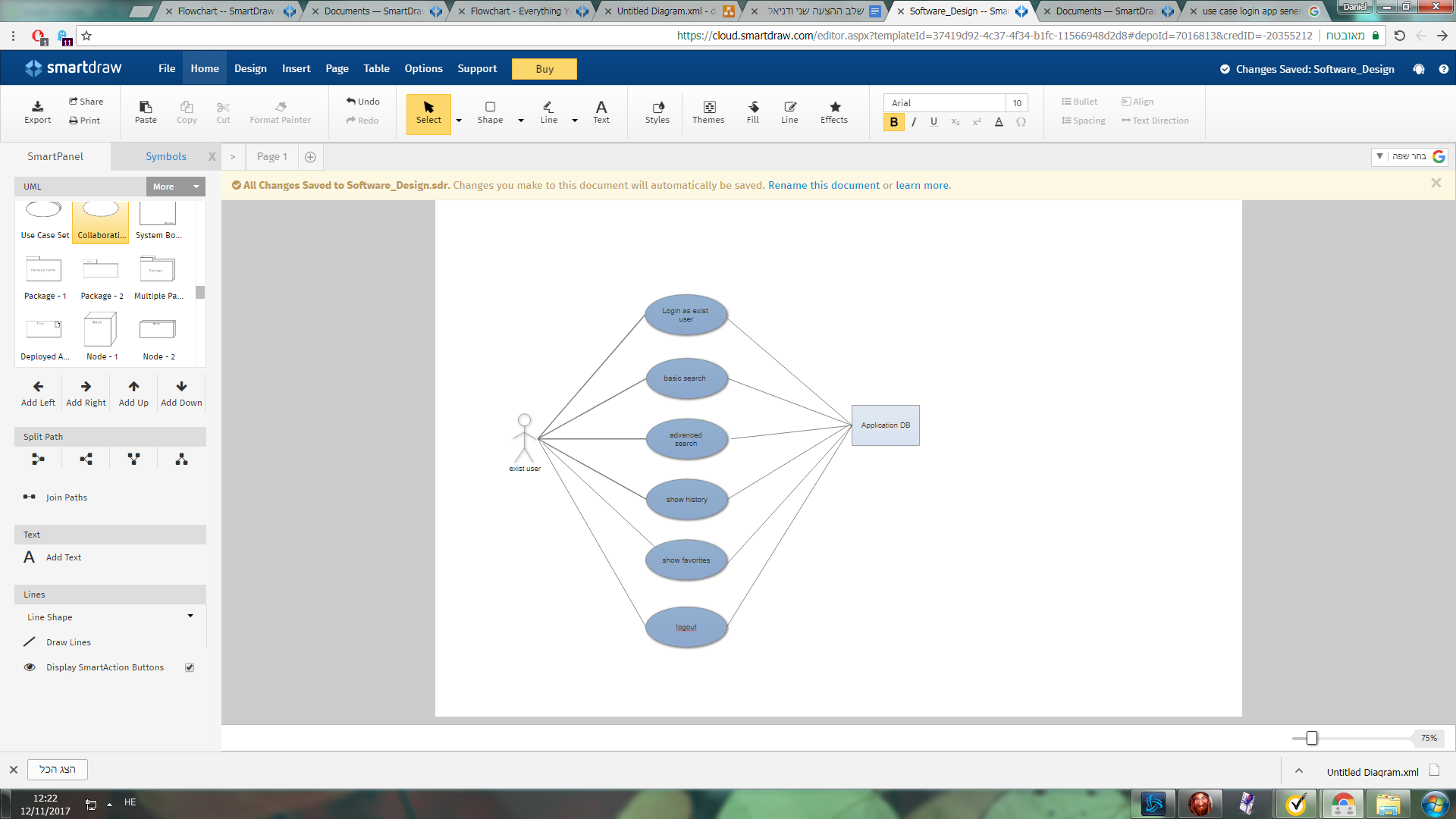
**UML CLASS DIAGRAM** - רעיון ראשוני לתרשים מחלקות - צד לקוח.



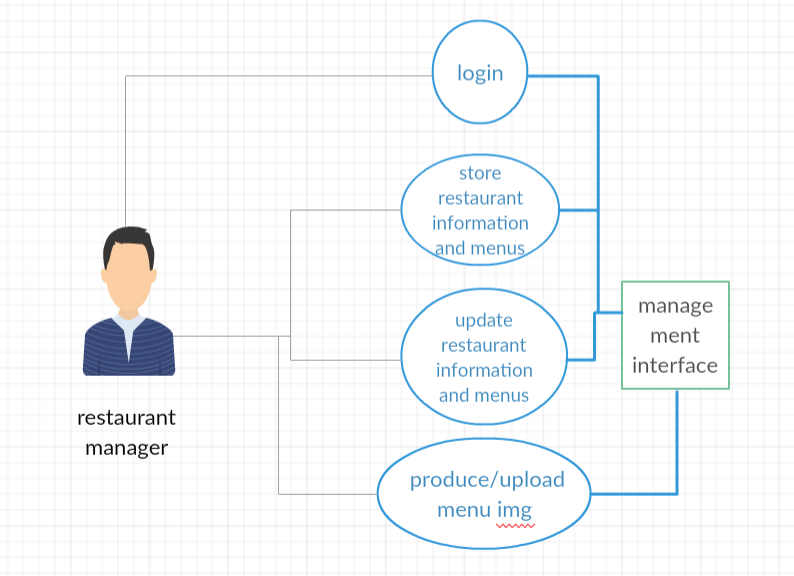
תרשים זרימה - בתרשים זה אנו נראה שימוש באפליקציה עבור משתמש קיים /אורח ואת פעולותיו .



תרשים **USE CASE** -בתרשים זה נראה את פעולות משתמש הקיים מול מסד הנתונים של האפליקציה משלב ההתחברות עד שלב התנתקות. במהלך החיבור יש לו אופציות שונות כגון: חיפוש בסיסי ומתקדם, צפייה בהיסטוריית החיפושים ובמקומות אשר סומנו כמועדפים.



בתרשים נראה את הפונקציונליות שיש לבעל המסעדה בממשק הניהול.



- תכנון הפרוייקט

|  |  |
| --- | --- |
| **3.11.17** | **פגישה ראשונית עם המנחה.** |
| **19.11.17** | **הגשת דוח הצעה למנחה.** |
| **סוף נובמבר 17** | **עמדת צד שרת עם תמיכה במסד נתונים ותקשורת ראשונית עם צד לקוח.**  **תחילת עבודה על המסכים בצד לקוח.** |
| **סוף דצמבר 17** | **כל המסכים בצד לקוח+פונקציונלית בסיסית בצד לקוח. עם APIים מתאימים בצד שרת.** |
| **ינואר 2018** | **סיום ממשק למילוי התפריט עבור בעלי המסעדות** |
| **ינואר 2018** | **יצירת קשר ראשוני עם בעלי מסעדות כדי לקבל תפרטים ושיווק ראשוני למוצר.** |
| **סוף סמסטר א** | **השקת אבטיפוס ראשוני.** |
|  |  |
|  |  |
|  |  |

## 

## **-** טבלת סיכונים

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **הסיכון** | **חומרה** | **מענה אפשרי** |
| **1** | **קושי הסתגלות לשפות ולפלטפורמות עבודה חדשות.** | **5** | **השקעה מרובה בשלב הלימוד של השפה. אלטרנטיבות שימוש נוחות זמינות.** |
| **2** | **עמידה בזמנים** | **5** | **חילוק ותכנית עבודה ללוז מתוכנן. לעמוד בו.** |
| **3** | **עבודה בצוות** | **2** | **הקשבה וכבוד הדדי בין חברי הצוות** |
| **4** | **עמידה בכל הפיצרים המתוכננים(חלקם מורכבים יותר)** | **3** | **להשקיע קודם כל בחלק המרכזי של הפרויקט, ורק אחר כל בכל השאר.** |

## 