**נספח 1 - (Software Requirements Document (SRD**

הקדמה

להלן מסמך דרישות המגדיר את הדרישות עליהן יש לענות להשגת מערכת עזר לניהול חניון. מערכת זו תהווה כלי העוזר לבעל חניון בקבלת החלטות ניהוליות שוטפות ותקופתיות, וכן תתן שרות לנהג המבקש להיכנס אל החניון.

הרעיון למערכת נולד מתוך בעיה הקיימת בתוך חניון מכללת אפקה. בחניון אפקה ישנה תופעה שמניית המכוניות היוצאות ונכנסות לחניון וממנו בשלב כלשהו סוטה מכמות המכוניות שנמצאות בחניון בפועל. בכך, לא ניתן להפיק הרבה מידע מנתוני המערכת ולנתחו לכדי מסקנות מהימנות. נוסף על בעיה זו קיימת בעיה נוספת בחניונים רבים. כאשר החניון מלא, מכוניות לא נכנסות אליו וממשיכות הלאה. זאת למרות שקיימת האפשרות שמיד לאחר שיעזבו את החניון בחיפוש אחר חניה אחרת, תצא מכונית מהחניון ותתפנה חניה לרווחתה של המכונית המבקשת לחנות.

על מנת לנסות לענות על בעיות אלו, מנוסח כאן כלל הדרישות ממערכת שתהווה כלי עזר ושירות לבעל החניון ולנהגי המכוניות.

תיאור המודל

מערכת StatistiPark נדרשת לעמוד ביעדים הבאים ומשמשת את השחקנים להלן:

**בעל החניון** – בעל החניון הינו שחקן ראשי ובעל עניין במערכת. המערכת מייצרת למענו סטטיסטיקות הנוגעות להחלטות היום-יומיות שעליו לקבל בחניון וכן החלטות תקופתיות שיעזרו לו בניהול חכם ויעיל יותר.

דוגמאות לסטטיסטיקות שתעמודנה לרשותו של בעל החניון:

1. מציאת איזורי חניה עמוסים יותר ופחות במסגרת זמן משתנה – לדוגמה, כאשר בעל החניון מעוניין לדעת אילו איזורים עליו לתחזק יותר בתוך החניון, הוא יוכל לדעת אילו איזורים היו יותר בשימוש ברבעון האחרון וכך להפנות יותר משאבים אל איזורים אילו שבאופן טבעי בעלי בלאי גבוה יותר.
2. מציאת זמני השימוש של חניה מסוימת.
3. מציאה של חניות הנמצאות בשימוש גבוה (בניגוד לאזורים שלמים הנמצאים בשימוש גבוה) – זאת על מנת לשקול להקצות אותן כחניות לנכים, שכן, הן כנראה חניות שמהן יהיה קל יותר לנכה להתנהל בחניון.

**מפעיל החניון** – מפעיל החניון הינו שחקן ראשי במערכת. המערכת מפותחת בשבילו ומתאימה את עצמה לצרכיו. היא תציג לו את הנתונים על המצב הנוכחי של החניון.

דוגמאות לנתונים שיעמדו לרשותו של מפעיל החניון:

1. מספר החניות בחניון.
2. מספר החניות התפוסות בחניון.
3. מספר החניות הפנויות בחניון.
4. מפה של החניון המספקת מידע באשר למיקומה של כל חניה ואם היא תפוסה או פנויה.
5. כאשר החניון מלא, מוצג למפעיל הזמן המשוער שייקח לחניה כלשהי בחניון להתפנות.

**נהג** – נהג המגיע אל החניון הינו בעל עניין במערכת. המערכת חוזה בתוך כמה זמן תתפנה חניה כאשר החניון מלא. משום כך, כאשר נהג בא להחנות את רכבו בחניון כאשר הוא מלא, המערכת תספק לו את הידע באשר לכמה זמן הוא יצטרך לחכות עד שתתפנה חניה בחניון.

הארכיטקטורה שנבחרה לפתרון הבעיה הנ"ל מתבססת על מערכת המורכבת ממספר מודולים. להלן דיאגרמת בלוקים המתארת מודולים אלה יחד עם הסבר על כל אחד מהם.

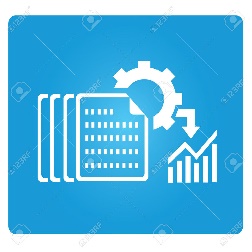


עיבוד נתונים

איסוף מידע

איסוף מידע  
(סימולטור)

מערך Database



הצגת נתונים למשתמש





תיאור תתי המערכות של מערכת StatistiPark

* איסוף מידע: מודול איסוף המידע אוסף נתונים על מצב החניון בעזרת חיישנים הפרוסים בו.
* מערך Database: מערך הdatabase- אחראי על שמירת כלל הנתונים המגיעים ממודול איסוף המידע ואחזורם עפ"י דרישת מודול עיבוד הנתונים.
* מודול עיבוד הנתונים: לב המערכת. מודול זה מבצע בפועל את האלגוריתמים המייצרים את הסטטיסטיקות והדו"חות השונים. בנוסף, מודול זה גם מבצע את אלגוריתם החיזוי.
* הצגת נתונים למשתמש: מודול המשמש כ-GUI עליו מוצגים המצב הנוכחי של החניון ופלטי מודול עיבוד הנתונים.

הגדרת דרישות

הדרישות ממערכת StatistiPark הינן מגוונות ונוגעות לסטטיסטיקות ודוחות שעל המערכת לספק.

* המערכת תכיל מסוף התממשקות דרכו יכולים המשתמשים השונים לתקשר איתה ע"י הזנת קלט וקבלת פלט.
* ביצוע חיזויים באשר להתפנותה של חניה בחניון כאשר הוא מלא.
* מציאת איזורי חניה עמוסים יותר ופחות.
* מציאת זמני השימוש של חניה מסוימת.
* מציאת חניות הנמצאות בשימוש גבוה.

להלן Use Case Diagram המציג את פעולת המערכת באמצעות זרימות שונות.

שם use case: **כניסה לחניון**

שחקנים משתתפים: נהג

זרימה מרכזית:

1. נהג נכנס אל החניון ומוצא חניה.
2. המערכת מעדכנת את נתוניה.

זרימה חלופית:

-אין-

שם use case: **הגעה לחניון מלא**

שחקנים משתתפים: נהג

זרימה מרכזית:

1. נהג מגיע אל הכניסה לחניון ומעוניין להחנות את רכבו.
2. בהגיעו, החניון מלא עד אפס מקום.
3. הנהג בודק בתוך כמה זמן תתפנה חניה בחניון.
4. הנהג מחליט לחכות עד להתפנות חניה בחניון.
5. חניה אחת מתפנה בחניון במסגרת הזמן האמור.
6. הנהג נכנס אל החניון ומחנה את רכבו.

זרימה חלופית א':

א1. בצעד 4, הנהג מחליט לא לחכות עד להתפנות חניה בחניון.  
א2. הנהג עוזב את המקום.

זרימה חלופית ב':

ב1. בצעד 5, אף חניה אינה מתפנה במסגרת הזמן האמור.  
 ב2. על אף זאת, הנהג מחליט לחכות עוד.  
 ב3. חניה אחת מתפנה בחניון.  
 ב4. הנהג נכנס אל החניון ומחנה את רכבו.

זרימה חלופית ג':

ג1. בצעד 5, אף חניה אינה מתפנה במסגרת הזמן האמור.  
 ג2. הנהג עוזב את המקום.

שם use case: **התעדכנות במצב החניון**

שחקנים משתתפים: מפעיל חניון

זרימה מרכזית:

1. משתמש מזדהה מול המערכת כמפעיל חניון.
2. המערכת מציגה את "מסך מפעיל חניון".
3. מפעיל החניון מקבל נתונים על מצבן של החניות בחניון ומיקומן.

זרימה חלופית:

-אין-

שם use case: **חילוץ נתוני מערכת**

שחקנים משתתפים: בעל חניון

זרימה מרכזית:

1. משתמש מזדהה מול המערכת כבעל חניון.
2. המערכת מציגה את "מסך בעל חניון".
3. בעל החניון בוחר בסטטיסטיקה/דו"ח הרצוי לו.
4. המערכת מציגה את הסטטיסטיקה/דו"ח הנבחר.
5. המערכת מציגה את "מסך בעל חניון".

זרימה חלופית:

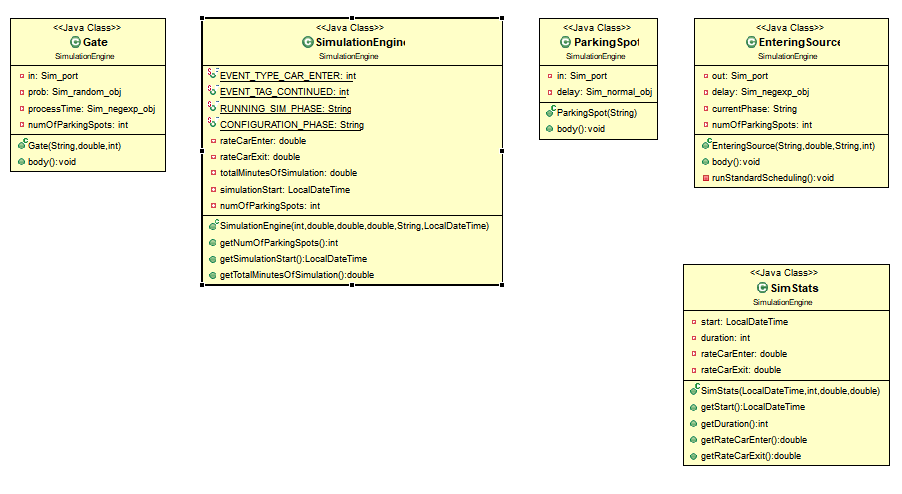
-אין-

אלגוריתם החיזוי המשמש לחיזוי הזמן שייקח לחנייה להתפנות בחניון כאשר הוא מלא.

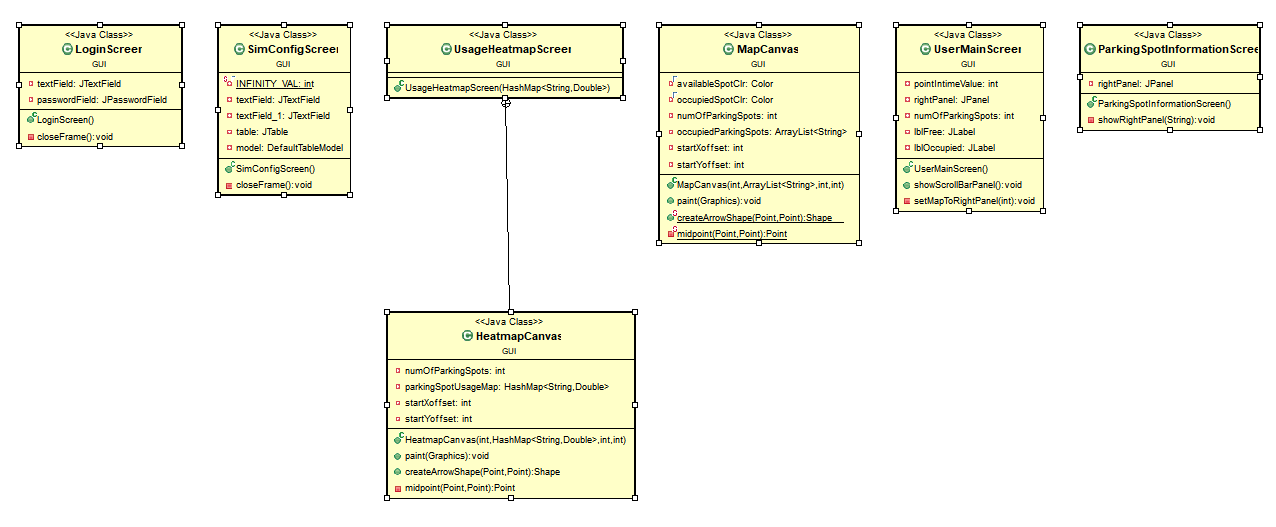
האלגוריתם כולל בתוכו דגימה של קצב הכניסות והיציאות באינטרוול (מקטע זמן) הנוכחי שבו מתבצע החיזוי (זאת על מנת לקיים את תכונות שרשרת מרקוב). לאחר מכן, האלגוריתם משקלל את הנתונים בתוך פונקציית החיזוי. דבר זה מתבצע מספר פעמים, עד להשגת אחוז דיוק של לפחות 85% ומחזיר את הזמן המינימלי שצריך לחכות על מנת שחניה תתפנה בהסתברות הנ"ל. אחוז זה עולה מתוך סקר ציפיות משתמש שנערך בקרב כמה עשרות משתמשים. להרחבה בסקר ניתן לפנות לדוח התכנון, פרק 5.

UML Class Diagram

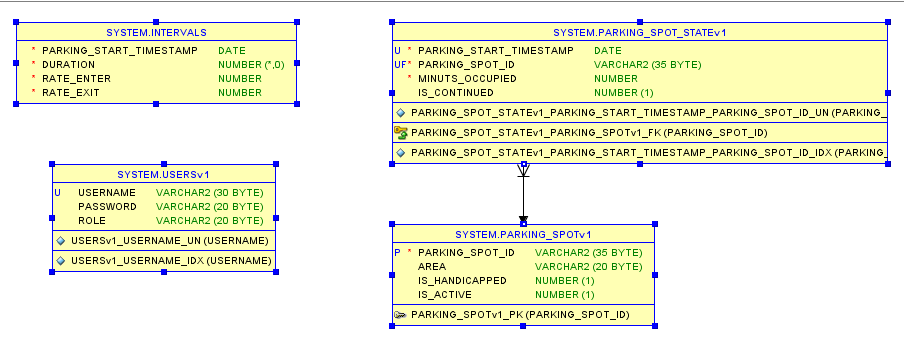
מודול הסימולטור



מודול ה-View



מבנה בסיס הנתונים



דרישות משאבים

דרישות הפעלה:

* דיסק קשיח עם 50GB פנויים.
* זיכרון RAM – 2GB.
* מערכת הפעלה Windows 7 ומעלה.
* בסיס נתונים Oracle SQL מותקן.
* שפת Java מותקנת.

דרישות אבטחה

המערכת נדרשת להיות מאובטחת מספיק על מנת לא לאפשר גניבה של נתוני המערכת הנאספים על ידה. יחד עם זאת, החשש מפני ניסיון גניבה שכזה הינו קטן למדי מאחר ומדובר בנתונים יבשים שאינם מסוגלים לשמש גורם חיצוני. על כן, על המערכת חלות דרישות האבטחה הבאות:

* הפרדת סוגי משתמשים וחלוקתם לשתי רמות: מפעיל חניון ומנהל.
* חשיפת מידע על פי רלוונטיות למשתמש המחובר. מידע יינתן על בסיס "צריך לדעת".
* אימות ברמה אפליקטיבית. זיהוי משתמשים בעת ההתחברות למערכת.
* אימות ברמת בסיס הנתונים. זיהוי משתמשים בעת ההתחברות לבסיס הנתונים.