**“Shift it easy”**

**מהו המוצר:** אפליקציית "לקוח שרת" שמטרתה להקל על מכין סידורי העבודה בעיקר ועל העובדים בנוסף.

**על איזה בעייה המוצר עונה:** ידוע שברוב מקומות העבודה בהם עובדים בשיטת משמרות הכנת סידור העבודה היא מטלה קשה, מסובכת ומסורבלת, צורכת הרבה זמן, הרבה נייר והרבה סבלנות.  
 “shift it easy” כשמו, בא להקל על הפעולה הזו, לתת למסדר העבודה כלי בעזרתו סידור העבודה יהיה מהיר יותר, איכותי יותר ובזבזני פחות.  
קיימות בשוק תוכנות שנותנות שירות דומה אך לא מאפשרות למסדר העבודה להפעיל שיקול דעת, כלומר הכנס אילוצים של עובד - קבל סידור מתאים.  
“shift it easy” נותנת למסדר העבודה את האופציה לשבץ את העובדים בסידור כראות עיניו תוך מתן משוב על צורת השיבוץ, כלומר האם הדרך בה החליט לשבץ תואמת את האילוצים של העובד שיוזנו טרם השיבוץ.  
כיום סידור העבודה מצריך גם מהעובדים לבוא במיוחד ולסמן את האילוצים שלהם לקראת הסידור הבא או לתת אילוצים בצורה מסורבלת ולא מרוכזת (בשיחת טלפון, בSMS וכו'...).  
“shift it easy” באה לפתור את הבעייה הזו ולייעל את הפעולה גם מצד העובדים, לתת לעובד להזין את אילוציו מכל מקום בכל שעה תוך מתן דד-ליין להזנת אילוצים, וגם כמובן שזה גם מקל על מסדר העבודה כי כל המידע מרוכז במקום אחד.

**מועמד ללקוח: אלישבע לוין, אחות ראשית (מסדרת המשמרות), במחלקת נשים של הדסה הר הצופים י-ם וצוות המחלקה (העובדים).**

המוצר כדאי לפיתוח הן מבחינת חסכון בזמן (זמן = כסף) והן מבחינת חסכון בנייר.  
כיום הכנת סידור עבודה לוקחת לפחות 4-5 שעות ונעשית באופן ידני על נייר,   
להערכתנו “shift it easy” יוריד את זמן הכנת הסידור משמעותית וכמובן יחסוך את השימוש בנייר, חבל לזרוק נייר על כל טעות, המערכת תתקן בשבילך.

**תיכון ראשוני - “shift it easy”**

|  |
| --- |
| Employee |
| constrainsList<Constrain> countOfRedConstrains  information (id,name…) |
| addConstrain(Constrain c) /\* add new constrain(s) \*/ save(); /\* save for now \*/  update(); /\* update constraints \*/  submit(); /\* send to the manager \*/ |

|  |
| --- |
| Constrain |
| shiftType /\* Morning, afternoon, evening \*/  Date type /\* for each shift choose: want to work, prefer not to, can’t \*/ |
|  |

**צד לקוח:**

|  |
| --- |
| Manager |
| employeesList<Employee> shiftsList<Shift> |
| legalInsert(); /\* check if the next insert is legal \*/ save(); /\* save the schedule \*/  update(); /\* update the schedule \*/ |

|  |
| --- |
| Shift |
| Name hours numberOfCrewMembers |
| enoughCrewMembers(); |

**צד שרת:**

**ממשק הפעלה ושפת תכנות מתכוננים:** עבודה עם python ואחסון המידע בשרת datastore של גוגל.  
שימוש ב- appEngine.  
  
**תיאור ממשק המשתמש המתוכנן**:

* **עובד –** דף הזנת מידע בסיסי.
* **מסדר עבודה –** דף ניהול המשמרות, מוצג ע"י לוח סידור לשבועיים הבאים ורשימת העובדים לצידו כך שמסדר העבודה יוכל לגרור עובד לתוך הלוח והמערכת תיתן או לא תיתן לא לשבץ את העובד במשבצת המשמרת בהתאם לאילוצים שהוזנו טרם הסידור.

**סיכונים:**

* חוסר שיתוף פעולה מצד העובדים.
* חוסר יכולת של מסדר העבודה להבין את המערכת.
* הסתבכות של העובדים עם המערכת.

**שלבי עבודה:**

* בניית סקיצת תרשים ראשונית (UML).
* חשיבה קבוצתית על מבנה ואופי הקוד.
* התייעצות על ממשק המשתמש תוך חשיבה על איך לבנות אותו בצורה הנוחה ביותר.
* חלוקת תפקידים בקבוצה.
* פגישות סנכרון.
* הצגת תוצר ראשוני ללקוח וקבלת משוב עבור התוצר.
* שיפור, התייעלות והמשך עבודה.

**צורת התמודדות עם הסיכונים:** כדי לפתור את הסיכונים הנ"ל, נבנה ממשק משתמש נוח שתואם את הצרכים של כולם ובעיקר מזמין, כך שהעבדים לא יראו סיבה לא להשתמש בו ומסדר העבודה לא יסתבך איתו.  
כמובן לאחר הכנה ראשונית של התוצר, מתן הממשק למשתמש ילווה בהסבר מפורט על אופי השימוש בו ועל יעילותו לשני הצדדים.