המחלקה להנדסת תוכנה

פרויקט גמר – תשע"ז

פיתוח שרת ואפליקציית משתמש

לניהול בית אופנה תמר פרץ

Cloud server & application

For Fashion Home Management

**בשיתוף**

מאת:

**שמרי פרץ**

**עמירם אלזם**

מנחה אקדמי: מר שי תבור

מערכות ניהול הפרויקט

|  |  |  |
| --- | --- | --- |
| # | מערכת | מיקום |
| 1 | מאגר קוד | https://github.com/shimriper/Fashion-Home-Menagement |
| 2 | יומן | https://github.com/shimriper/Fashion-Home-Menagement/wiki/Job-Log-(diary) |

* תקציר
* תוכן העניינים

(שלב זה יושלם בהמשך)

# **תקציר:**

בפרוייקט זה נציג אפליקציית web למעצבת שמלות כלה וערב תמר פרץ, בעלת ניסיון של עשרות שנים בתחום, אתר זה פותח לבקשת תמר פרץ, על מנת לסדר ולהקל על ניהול העסק בהיבט הכלכלי והתפעולי.

מטרתו העיקרית של האתר היא מערכת חכמה שתעשה סדר בניהול ותפעול העסק, לפיכך קיים כניסה למערכת על פי תפקידים:

* **תופרת – Seamstress:**

למשתמש הזה תהייה גישה לפרטים של השמלה, על מנת שהשמלה תהייה בדיוק כפי שהלקוחה ביקשה ובזמן שהיא צריכה.

* **מנהל הזמנות –Reservation Manager :**

מנהל ההזמנות הוא משתמש שיושב עם הלקוחה ולוקח את כל הפרטים של הלקוחה ושל השמלה שהיא רוצה, כולל פרטים אישיים, מידות, צבעים וכו', ומכניס אותם למערכת להמשך טיפול התופרת.

* **מנהל – Director**:

משתמש זה הוא המשתמש בעל הגישה לכל התחומים בעסק. משתמש זה יוכל לראות את כל ההזמנות שקיימות במערכת, את פרטי ההזמנה וכמובן את סטטוס ההזמנה.

בנוסף לכך, לשלושת המשתמשים השונים קיים דרך לתקשר ביניהם על ידי שליחת הודעות בתוך המערכת, על מנת שיוכלו לעדכן ולהתעדכן בפרטי ההזמנה.

1. מבוא

בית אופנה תמר פרץ – מייצר שמלות כלה וערב.

בפרויקט ננהל מערכת הזמנות של לקוחות בסטודיו לשמלות כלה וערב.

המערכת תכלול את כל השלבים בתהליך עד לקבלת המוצר בשלמותו.

המערכת תארגן את התהליך כך שכל הפועלים ידעו מה המצב של ההזמנה בכל רגע נתון, ומה השלב הבא בתהליך ומתי צריך לסיימו.

בנוסף נוכל לראות את קצב התקדמות התשלומים של המוצר.

המערכת תעזור בניהול הכללי של העסק ותייצא דוחות, דבר שיעזור להעריך את המצב הכללי בעסק.

כיום כל המידע נשמר בצורה לא יעילה על דפים וקבצי EXEL שקשה מאוד לעבוד איתם.

1. תיאור הבעיה

בית אופנה "תמר פרץ" עובד כיום עם טפסים וקבצי WORD ו EXEL וניירות.

רוב המידע נכתב ידנית בצורה לא יעילה ולא מאורגנת.

כיום קשה מאוד לדעת מה מצב של כל שמלה בצורה גלויה, שלבי ההתקדמות של השמלות לא נרשמים.

מבחינה כלכלית – העסק מגלגל כספים בכל יום וקשה מאוד לעקוב אחר ההוצאות וההכנסות כיוון שכל המידע נרשם על דפים, ולאחר מכן מוזן לתוך EXEL או WORD.

דבר שמבזבז המון זמן וכוח עבודה.

מבחינה ניהולית – קשה מאוד כיום לבצע את הניהול של ההזמנות בכל המובנים. למשל – ברגע שנכנסת הזמנה עד התוצר הסופי יש מספר שלבים שעוברת השמלה.

בכל שלב צריך לדעת מה המצב של השמלה כדי לדעת אם אנו עומדים בזמנים של אותם השלבים ובכלל. כי לכל שמלה יש תאריך יעד – תאריך חתונה / תור למדידה.

כיום אין ניהול של הכספים לפי ההתקדמות של השמלה – ברגע שהשמלה עומדת בשלב מסוים צריך לדרוש כסף על אותו השלב, היום עושים זאת ללא התראות ובצורה שבא אפשר לשכוח מתשלום על שלב.

דרישות ואפיון הבעיה

* 1. במערכת יהיו 3 רמות של משתמשים:
  2. מנכל – מסך שבו יראו את מצב ההזמנות.
  3. מנהל הזמנות – מסך של הזנת נתונים על ההזמנות.
  4. תופרות – מסך של מה נדרש לעשות בכל שלב.

הבעיה מבחינת הנדסת תוכנה

בפרויקט זה אנו צריכים ליצור סוג של דיבור והבנה בין כל המשתמשים במערכת.

כך שכל משתמש יבין את המצב הכללי של השמלה, ומה נדרש ממנו לעשות.

האתגרים:

1. ממשק נוח לתפעול – כיוון שאנשים אלו אינם משתמשים במחשב ביום יום.

דבר שידרוש תשומת לב רבה, כדי שהמערכת תהיה פשוטה ככל האפשר,

קלה להבנה וקלה לתפעול.

1. כתיבת שרת צריכה להיות פשוטה – בכדי להוסיף שינויים או תוספות בהמשך.
2. שמירת המידע – איך לשמור את המידע, יש המון מידע שישמר דבר שנצטרך להתחשב בו.(כרגע מחפשים db חינמי אם קיבולת גבוהה)
3. תיאור הפתרון

באופן כללי – אנו נפתח את המערכת עם הלקוח ורצונותיו כדי שיבין את המערכת, וכדי שהמערכת תהיה מובנת לו ועונה על צרכיו.

תהליך זרימת המידע:

משתמש מנהל הזמנות

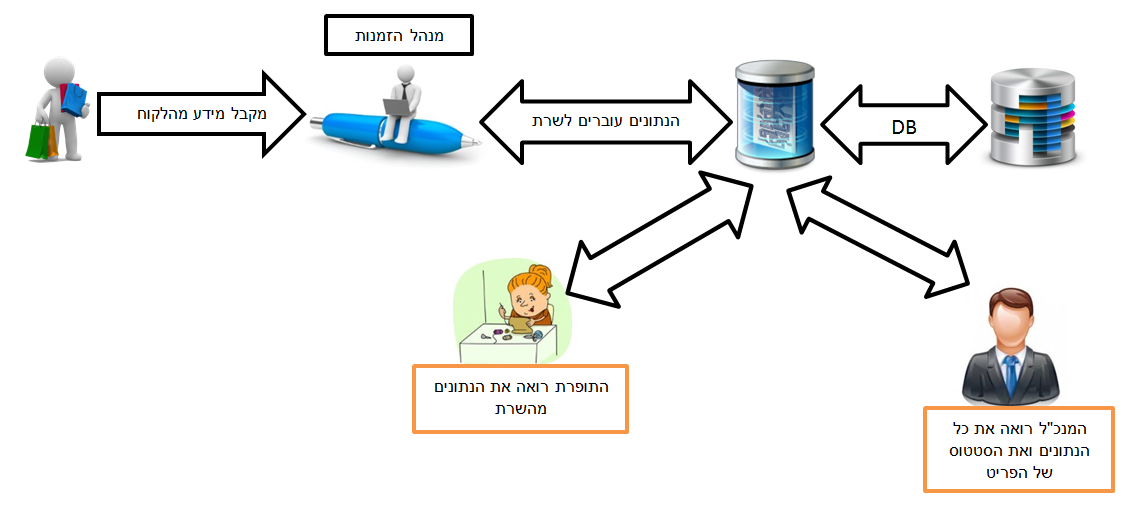
1. מזין את הנתונים על הלקוח.
2. המידע נשלח לשרת.
3. מהשרת ל DB

משתמש תופרת

1. מקבל את המידע במערכת על הלקוחות.
2. בוחר מרשימה את הלוקח שצריך לעבוד על המוצר שלו.
3. המידע מועבר מהשרת לclient .
4. עדכון השלב שהושלם.
5. שליחת המידע לעיבוד של השרת
6. שמירת המידע העדכני ב DB.

משתמש מנכ"ל - צפייה בכל המשתמשים הפועלים.

מהי המערכת

****התרשים מתאר את המערכת, מנהל ההזמנות לוקח את ההזמנה מהלקוחה והנתונים מועברים ל-DB , המשתמשים האחרים מקבלים הודעה שישנה הזמנה חדשה ומושכים את הנתונים מהשרת. כל משתמש רואה את הנתונים הרלוונטים בשבילו. התופרת רואה את סוג השמלה, סוג הבד, מידות ותאריך סיום. משתמש מנהל רואה את כל הפרטים שמנהל ההזמנות לקח מהלקוחה ואת סטטוס ההזמנה.

רכיבי המערכת:

1. בסיס נתונים mongodb
2. שרת – Nodejs express
3. פיתוח ממשק web – לצורך ניהול ובקרה.

**תהליכים ונתוני המערכת:**

המערכת כוללת שלושה מצבים של עבודה:

מצב א' – מנהל הזמנות, מצב שבו מנהל ההזמנות לוקח את פרטי הלקוחה והשמלה שהיא רוצה ושולח את הנתונים לשרת.

מצב ב' – תופרת, מצב שבו התופרת רואה את הנתונים של השמלה ותופרת אותה על פי הנתונים שהוזנו לשרת, לאחר סיום שלב בשמלה התופרת מעדכנת את המערכת על סטטוס השמלה.

מצב ג' – מנהל, מצב שבו המנהל רואה את ההזמנות והתשלומים שהתקבלו ומעדכן את השאר בהמשך טיפול.

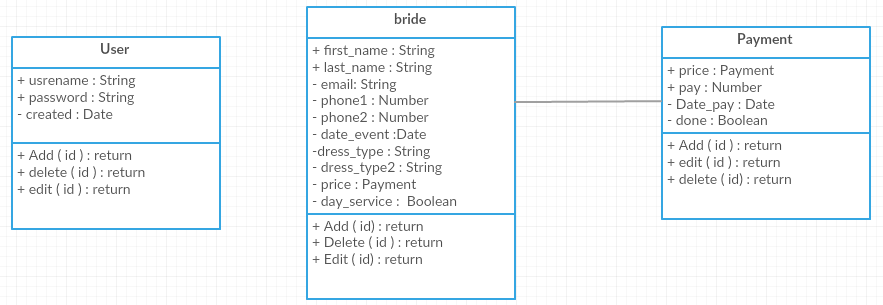
**תיאור הכלים המשתמשים לפתרון:**

בפרוייקט זה נשתמש בסביבת עבודה code visual studio.

דפי האתר נכתוב בשפות html ,bootstrap, angular.

צד שרת נכתוב ב- nodejs ובסיס הנתונים יהיה mongoose on mongoDB.

לניהול בסיסי הנתונים נשתמש ב- mLab.

**דיאגרמת UML:**

1. נספחים
   1. רשימת ספרות / בביבליוגרפיה.
   2. תרשימים וטבלאות.

**תכנית בדיקות:**

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **דף** | **הבדיקה** | **מה נקבל** |
| **1** | התחברות | הכנסת אימייל בלבד | הודעת שגיאה |
| **2** | התחברות | הכנסת סיסמא בלבד | הודעת שגיאה |
| **3** | התחברות | הכנסת כל הנתונים | התחברות למערכת |
| **4** | הזמנה | שדות החובה ללא ערכים | הודעת שגיאה |
| **5** | הזמנה | הכנסת כל הנתונים | יצירת collection חדש בDB |
| **6** | הזמנה | הכנסת כל הנתונים | יצירת הזמנה חדשה |
| **7** | כל הכלות | לחיצה על הלקוח | פתיחת פרטי ההזמנה |
| **8** | הודעות | שליחה וקבלת הודעות | שליחה וקבלת הודעות |
| **9** | תשלומים | עדכון תשלום | יתרת התשלום |

**תכנון הפרוייקט:**

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **תאריך** | **פגישה** | **הערה** |
| **1** | 20/10/2016 | פגישת הכרות | הבנת הבעיה - חשיבה על המערכת |
| **2** | 28/10/2016 | אפיון ראשוני של המערכת | תאום ציפיות |
| **3** | 20/11/2016 | שלב ההצעה | עזרה מהלקוח לשלב זה |
| **4** | 21/11/2016 | תחילת העבודה | חלוקת משימות |
| **5** | 19/12/2017 | יצירת מאפיינים בשרת | MongoDB |
| **6** | 11/1/2017 | סיום מסכים ראשונים | מסכי כניסה והזמנות |
| **7** | 12/1/2017 | פגישה עם הלקוח | הצגה ראשונית והמשך עבודה |
| **8** | 22/1/2017 | מסכי התשלום | על פי בקשת הלקוח |
| **9** | 26/1/2017 | סיום מסכי התשלום | על פי בקשת הלקוח |

טבלת סיכונים

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **סיכון** | **חומרה** | **מענה אפשרי** |
| **1** | סנכרון בין הגורמים | קריטי | טבלת מעקב - חלוקת משימות |
| **2** | אי עמידה בלו"ז | קריטי | טבלה עם תאריך יעד |
| **3** | פערים טכנולוגים בהפעלת האלמנטים | בינוני | למידה מעמיקה וניסויים לפני בניה |
| **4** | שרת בעל תעבורה לא מספיקה | בינוני | בדיקת שרת חינמי טוב יחסית |

* 1. רשימת / טבלת דרישות

טבלת דרישות (User Requirement Document)

|  |  |  |
| --- | --- | --- |
| # | תיאור דרישה | עדיפות 1-10(1 נמוך 10 גבוהה) |
| 1 | בניית שרת | 10 |
| 2 | בניית DB | 10 |
| 3 | בניית אפליקציית משתמש | 7 |
| 3 | עיצוב האפליקציה | 5 |