Azrieli
College of Engineering
Jerusalem
עזריאלי
מכללה אקדמית להנדסה
ירושלים

המחלקה להנדסת תוכנה

פרויקט גמר – תשע"ו

שבוס - טרמינל לאוטובוס

Shabus Terminal Application

מאת

אברהים אבו רמילה

206959926

Ramat Bet Hakerem, P.O.B 3566, Jerusalem 9103501 רמת בית הכרם, ת.ד. 3566, ירושלים מיקוד
Tel. +972-2-6588000 | Fax. +972-2-6588001 | Email: info@jce.ac.il | Web: www.jce.ac.il

**** יש לציין כי פרויקט זה מתבצע בזוג (עם מוראד חליל 315763441) מאחר וגודלו (בינוני-גדול) מצריך מאמץ של שני סטודנטים . הפרויקט מקיף נושאים רבים בכיוונים שונים . העבודה מתחלקת לפי סבבים, בכל סבב תיעשה חלוקת משימות שווה בשווה הן תכנותיות והן ניהוליות.**

מנחה אקדמי: מר שי תבור אישור: ShayTavor תאריך:19/11/2017

אחראי תעשייתי: גב' נדב כהן אישור: NadavCohen תאריך: 8/11/2017

רכז הפרויקטים: דר' אסף שפנייר אישור: תאריך:

מערכות ניהול הפרויקט:

#	מערכת	מיקום
1	מאגר קוד	https://github.com/murradkh/Shabus-Teminal-App
2	יומן	https://www.my-diary.org/read/d/929360
3	בדיקות יחידה	http://getliner.com/mgbkC
4	אופן ביצוע הבדיקה	http://getliner.com/Oo33J
5	סרטון גרסת אלפא	https://drive.google.com/file/d/10A9J5yO-slbT4fTQUmSepiG4IX4-vmrz/view?usp=sharing

תקציר:

הפרוייקט שלנו הינו פרוייקט תעשייתי שנעשה בחסות האגודה השיתושפית "שבוס" שמספקת תחבורה שיתופית בשבתות . בשיתוף פעולה עם האחראי התעשייתי "נדב כהן" והמנחה האקדמי "שי תבור" . הפרוייקט בעיקר משמש לנהגי/נוסעי האגודה , הוא משיג להם הקלות והטבות אודות הנסיעה והחברות שלהם באגודה . המצב הנוכחי והניהול הידני של תהליכי/שירותי האגודה אינו מייעל/מעודד את הנסיעה. הדבר שמעלה את הצורך בניהול יותר יעיל ומאפשר חדשנות בתחום שירותי האגודה . בדיוק בנקודה זו אנו נכנסים לתמונה , אפליקציית מובייל שתנהל את הנסיעה הינו הפתרון הטוב והיעיל ביותר . האפליקצייה תשמש כ- "מסופון" – "טרמינל" בכניסה לאוטובוס, ומשם הלאה היא תציע בפני המשתמשים מגוון רחב של הטבות ושירותים.

תוכן העניינים :

1. מבוא
2. תיאור הבעייה
3. תיאור הפתרון
4. סקר שוק + סקירת עבודות קודמות
5. נספחים

6. תוכנית בדיקות
7. מסקנות עד פה , דברים שחשוב לציין להמשך הדרך .

מילון מונחים , סימנים וקיצורים:

תחב"צ = תחבורה ציבורית
הארגון = שבוס
העמותה = שבוס
האגודה = שבוס
הלקוח = "נדב כהן" מנכ"ל שבוס
שירותים = הטבות = דרישות הלקוח = פיצ'רים

1.מבוא

1.1 החשיבות של תחבורה ציבורית בישראל

אנו קוראים כתבות שהתחב"צ בישראל לא מספיק יעילה, ודורשת שיפור מתמיד. עולה כמובן שאלה חשובה מאוד- בעידן שבו ממילא יותר ויותר אנשים רוצים לרכוש רכב פרטי, האם יש עוד חשיבות בכלל לתחב"צ ? כפי שנראה בהמשך , זה ממש לא נושא שאפשר להקל בו ראש - ומדובר על נושא בעל חשיבות עליונה.

2.1 כל הסיבות לתחבורה ציבורית יעילה

עלות נמוכה- אי אפשר להשוות בין העלות של תחבורה ציבורית, לבין העלות של אחזקת רכב פרטי. לשם השוואה, המחירים של הרכבים הפרטיים מתחילים בכמה עשרות אלפי שקלים, ומכאן רק מטפסים למעלה. הוסיפו לכך עלויות של ביטוח, וכמובן
עלויות הדלק - שבימינו מגיע למחירים אסטרונומיים. יוצא מכך שגם אם לא רכשתם רכב חדש, אתם תוציאו מידי שנה כמעט 10 אלף ש"ח על אחזקת הרכב, וזאת בהנחה שלא עשיתם תאונה, שאין צורך לטפל ברכב, וללא העלות של טסט.
לעומת זאת, נסיעה קבועה באמצעי תחבורה ציבורית, יובילו אתכם להוצאה מקסימלית של -1000 2000 ש"ח בשנה בלבד, בהנחה שאתם משתמשים בתחבורה בכל יום לפחות פעמיים.

איכות הסביבה -שוחרי איכות הסביבה מציינים פעמים רבות את הקשר החיובי בין שימוש באמצעי תחבורה ציבורית, לבין שמירה על הסביבה. ראשית, אמצעי התחבורה הציבורית החדשים נחשבים מזהמים פחות מרכבים פרטיים. מעבר לכך, כלי רכב אחד של תחבורה ציבורית, שאולי מזהם כמו רכב פרטי - יכול להכיל כמות הרבה יותר גדולה של נוסעים. כך, אנשים יכולים להגיע ליעד בזמן, ועם זאת הסביבה נותרת פחות מזוהמת.

בטיחות - אנו קוראים לצערנו מידי יום כמעט, על תאונות דרכים שמתרחשות. במקרה הטוב הדבר מסתיים בנזק לרכוש בלבד. במקרים רעים יותר בפצועים, וכמובן שמידי שנה יש גם מאות הרוגים בתאונות. לא פעם דבר נגרם מנהגים שנסעו יותר מידי שעות על הכביש, מנהגים שנסעו שיכורים, ומעוד שפע של סיבות. לעומת זאת, נהגי התחבורה הציבורית הם מיומנים פי כמה, ואחוז התאונות שבהן מעורב כלי רכב ציבורי הוא אפסי כמעט. הנסיעה היא בטוחה פי כמה, ומונעת מצב שבו אדם צריך לנהוג כשמצבו הבריאותי, הפיזי או הנפשי לא תקין לגמרי.

אין צורך ללמוד - עוד יתרון של התחבורה הציבורית, הוא שאין צורך להוציא רישיון כדי לעשות בה שימוש. יש לזכור שלא כל אדם מרגיש בנוח במהלך מבחני הטסט להוצאת הרישיון. כמו כן - עלויות השיעורים הן גבוהות מאד, ולא כולם יכולים להרשות זאת לעצמם. שימוש בתחבורה ציבורית, מייתר את התהליך.

3.1 תחב"צ בירושלים

מערך התח"צ הישן בירושלים היה פרי התפתחות הדרגתית ארוכת שנים. התפתחות זו של העיר ושל אמצעי התחבורה הביאה ליצירת רשת קווי אוטובוס ארוכים ומסורבלים המחברים בין תחנות מוצא ליעדים מרוחקים, חוצים את מרכז העיר, ובחלקם משרתים רק אוכלוסייה קטנה של נוסעים. כדי ליצור מערכת פשוטה ויעילה יותר, תכנן משרד התחבורה רשת קווים מרכזיים וקווים המזינים אותם, כמקובל במערכי תחבורה מתקדמים בעולם. התכנון נועד להגביר את השימוש בתח"צ באמצעות קיצור משך הנסיעה, ולהעלות את איכות החיים במרכז ירושלים באמצעות הפחתת זיהום האוויר והרעש, הגברת הנגישות לבתי העסק ולמקומות הבילוי והגברת אמינות השירות. כל אלה תוכננו לגרום למשתמשים ברכבם הפרטי להעדיף את התחבורה הציבורית במרכז העיר.

באוגוסט 1997 אישרה הממשלה (בהחלטה מס' 2457) מדיניות ארצית להתמודדות עם בעיית העומס בכבישים, ובכלל זה בעיר ירושלים. בהחלטה נקבע כי תינתן עדיפות לתח"צ, וכי עד יולי 1998 יתוכננו מחדש קווי האוטובוס על פי רשת חדשה של נתיבים לתח"צ. בהחלטה גם נקבע כי בצמתים מרומזרים תינתן עדיפות לקווי אוטובוס, וכי עד סוף דצמבר 2000 תושלם סלילת הנת"צים המתוכננים.

4.1 תחב"צ בירושלים בשבת

בשנת 1951 נחקק "חוק שעות עבודה ומנוחה", האוסר על העסקת יהודים בשבת.

סוגיית השבת השנויה במחלוקת מבטאת את הניגודים העמוקים בין הדתיים לחילוניים. מרבית החילונים תומכים לכל הפחות בפתיחת מקומות בילוי ובהפעלת תחבורה ציבורית בשבת. אולם, מרבית הדתיים רואים חילול שבת בפרהסיה כפוגע בצביון היהודי של המדינה. בתחום השבת קבע מכתב הסטטוס קוו כי "ברור שיום המנוחה החוקי במדינה היהודית יהיה יום השבת, כמובן מתוך מתן רשות לנוצרים ובעלי דת אחרת לשבות ביום החג השבועי שלהם". המערכת הפוליטית נוהגת בסוגיית השבת בדפוסים מובהקים של דמוקרטיה הסדרית, כשהיא מתאפיינת בהתייחסות אגבית למעמד השבת בחקיקה הראשית, וכן בהאצלת סמכויותיה. לעצירת שחיקה הדרגתית בסטטוס קוו, נעשו ניסיונות, בעיקר מצד המפלגות הדתיות, לחוקק חוק שבת ארצי.

בתחום התחבורה הציבורית התבססו ההסדרים על הסטטוס קוו שהיה קיים ערב הקמת המדינה. ההבנות הפוליטיות בנושא לא נשענו על חוק כתוב אם כי על הסכמים בלתי רשמיים. בשכונות דתיות נאסרה כל תנועה של כלי רכב ומגוון רחב של תחומים נותרו בשטח האפור בהתאם לקביעה במסמך.

בישראל רוב התחבורה הציבורית אינה פעילה בשבת, פרט לקווי שירות המשרתים יישובים מרוחקים מאוד בצפון או בדרום, כדוגמת קווי אילת הפנימיים והחיצוניים, קווים המשרתים את ערי הצפון הרחוקות עד לקריית שמונה ועוד. בנוסף פועלת תחבורה ציבורית בעיר חיפה כולל קווים פנימיים עקב כך שלפני הקמת המדינה התחבורה בה פעלה גם בשבת וכך נשמר הסטטוס קוו. גם בערים מעורבות נוספות כמו נצרת ונצרת עילית פועלת תחבורה ציבורית בשבת. כמו כן,מוניות ומוניות שירות מורשות לנסוע בשבת.

בשנים האחרונות התעוררו מספר יוזמות פרטיות ששמו להן למטרה לספק שירותי תחבורה בסופי השבוע, ובפרט בשבת, באמצעות הקמת אגודות שיתופיות. במאי 2015 נחנך בירושלים מיזם התחבורה השיתופית "שבוס" (הלחם של שבת ואוטובוס), והחל להפעיל שירות במסלול ודמי נסיעה קבועים, שבו ניתן להשתמש לאחר רישום קצר לאגודה ותשלום דמי חבר שנתיים. בשנת 2017 התרחבו המיזם לערים נוספות, ונכון לקיץ אותה שנה מפעיל מיזם "שבוס" קו בירושלים ומעלה אדומים וכן קווים מירושלים, ראש העין וחולון לתל אביב.

2. תיאור הבעיה

1.2 קצת על הסביבה ה"בעייתית" – ירושלים

הבירה הכבירה: ירושלים היא העיר הגדולה בישראל ומתגוררים בה קצת יותר מ-815 אלף בני אדם. כך עולה מנתונים שפרסמה הלשכה המרכזית לסטטיסטיקה לרגל יום ירושלים. 515,200 מהתושבים הם יהודים ואחרים ו-301,100 - המהווים 37% מתושבי העיר - הם ערבים. במהלך שנת 2012 גדלה אוכלוסיית העיר ב-12,400 תושבים, בעיקר בשל ריבוי טבעי.

מבין היהודים בירושלים, 35% מגדירים את עצמם חרדים. 18% מגדירים עצמם דתיים, 12% מסורתיים-דתיים, 14% מסורתיים לא כל כך דתיים ו-20% חילונים. היתר, כאחוז אחד, לא הגדירו את עצמם. ביתר היישובים, שבעה אחוזים מגדירים את עצמם חרדים, עדרה אחוז כדתיים, 14% מסורתיים-דתיים, 24% מסורתיים לא כל-כך דתיים ו-45% חילונים.

מתושבי ירושלים (40% מהיהודים ו-34% מהערבים) מרוצים מהתחבורה הציבורית

באזור מגוריהם, בדומה לאחוז המרוצים בשאר היישובים. 38% מתושבי ירושלים מרוצים ממצב הכבישים והמדרכות באזור מגוריהם (43% מהיהודים ו-30% מהערבים), לעומת 57% מרוצים בשאר היישובים. ל-45% מתושבי ירושלים מפריע זיהום האוויר באזור המגורים (ל-36% מהיהודים ול-61% מהערבים), לעומת 39% בשאר היישובים בארץ. 70% מתושבי ירושלים מרגישים בטוח ללכת לבד בשעות החשכה באזור מגוריהם (71% מהיהודים ו-67% מהערבים), בדומה ל-72% בקרב תושבי שאר היישובים בארץ

בניתן הסטטיסטיקות הנ"ל , ניתן לראות שהרוב המוחלט של תושבי ירושלים הם חילונים. דבר זה מעיד ש-לאחוז הגדול של תושבי העיר אין הגבלה מטעם הדת לנסוע בשבתות. ובכך, אחוז קטן של תושבי העיר כופה אי-נוחות , אי-זמינות . ובמרקים רבים סכנת חיים , איך לא כשייתכן שתושב כלשהוא יצטרך להגיע לבתי חולים בשבת . כשאין בבעלותו רכב פרטי.

2.2 מה זה שבוס

שבוס הוא הסעה שיתופית שמחברת ירושלמים למוקדי הבילוי בסופי השבוע.
זהו מיזם של האגודה לתחבורה שיתופית בירושלים. האגודה היא קואופרטיב - התארגנות של חברים להסעות שיתופיות בסופי השבוע, במחירים שווים לכל נפש. שבוס יפעל בשעות קבועות על פי דרישות החברים ובתחנות איסוף שנקבעו מראש.
האגודה פועלת כמוסד ללא מטרות רווח. היא הוקמה על ידי פעילים חברתיים שגרים בעיר, אוהבים אותה ורוצים להפוך אותה למקום טוב יותר לחיות בו. הרעיון שלהם נולד כתוצאה מהפער הגדל והולך בין שפע מקומות הבילוי המשגשגים בשנים האחרונות בירושלים, לבין חוסר הנגישות שלנו אליהם בסופי השבוע. אנחנו רוצים לשנות את המצב הנוכחי שבו רק בעלי אמצעים יכולים ליהנות מניידות חופשית בסוף השבוע. אנחנו רוצים לשנות את המצב הנוכחי בו צעירים לוקחים את ההגה לידיים אחרי ששתו כי אין אלטרנטיבה במחירים סבירים. אנחנו רוצים להפוך את התרבות בעיר לשוויונית יותר, נגישה יותר ובטוחה יותר. אנחנו רוצים ליצור קהילה שנוסעת ביחד ולהפוך את התחבורה הפרטית בשבת לשיתופית יותר.
ההסעות -- שבוס -- נועדו לחברים בלבד. שבוס מציע לכם להצטרף כחברים בקואופרטיב וליהנות מההסעות. המסלולים, התדירויות, עלויות התפעול ודרך ההתנהלות ייקבעו כולם בהתאם לצרכים של החברים וליכולות של הקואופרטיב – על ידי החברים בו.

3.2 המצב כיום אינו אידאלי

המצב כיום , כלומר בלי מערכת ה "טרמינל" שתוצמד לאוטובוסי שבוס , אינו מעודד נסיעה בשבת, אפילו לא בכל יום אחר .
מדובר בהתנהלות לא יעילה אודות הנסיעות , איך לא כשכל הרישומים וההזמנות נעשות באופן ידני בעמותה . הדבר שמעכב את הנסיעה ומייגע את הנוסעים , ויגרום להם לשקול לרכוש רכב פרטי . ובכך חוזרים לאותה בעיה ולמעשה לא הרווחנו כלום !

איך הפרוייקט שלנו נכנס לתמונה ?

הטרמינל עומד לפתור את אי הנוחות הנ"ל ולהפוך את האוטובוס ל " חכם " יותר ואת הנסיעה ל-"חביבה" יותר . ע"י זה שהוא מאפשר כל מיני שירותים שדואגים לכך, כגון זיהוי נוסע/נהג, חישוב מסלולים וזמני הגעה של האוטובוסים , דיווח בזמן אמת , מחיר נסיעה לפי מרחקה וכו....

4.2דרישות ואפיון הבעיה

מטרת הפרויקט היא ליצור אפליקציה אשר תשמש (כמסופון)טרמינל לאוטובוס/ים אשר מפעילה שבוס בסופי שבוע. כיום לארגון יש אפליקציה WEB אך הארגון רוצה להרחיב את הפונקציונליות שלו וטכנולוגיה זו אינה מאפשרת זאת. כחלק מהתגברות על המגבלות הטכנולוגיות ניסה הארגון להשתמש בפתרונות צד שלישי אך גם זו אינה מספקת לו את הפונקציונליות הנדרשת לו.

המוצר שלנו בא לתת מענה לכל הדרישות הטכנולוגיות של הארגון.

דרישות (שירותים) עיקריים שהפרויקט מספק :

1. זיהוי הנהג – הנהגים בשבוס רשומים במסד נתונים, בתחילת המשמרת נהג יזדהה בעמוד הזיהוי באפלקיציה.
2. זיהוי נוסעים – על מנת להשתמש בשירותי שבוס יש להיות רשום לאגודה (הרישום נעשה באתר האגודה) כאשר הנוסע עולה לאוטובוס הוא יזין באפליקציה שלנו אמצעי זיהוי (מספר טלפון ע"פ דרישה של הארגון) והאפליקציה תזהה אותו מול המסד נתונים של שבוס, כלומר אם הוא רשום באגודה. בנוסף נוסע רשום יכול להעלות עימו עוד 4 נוסעים שלא רשומים באגודה. לאחר הזיהוי האפליקציה תתן לו אפשרות לבחור כמה נוסעים הוא מעלה איתו.
3. דיווח מיקום האוטובוס בזמן אמת – האפליקציה תשלח דיווח כל זמן נתון על מיקום האוטובוס על ידי רכיב ה GPS במכשיר.

**אלה רק טעימה קטנה מהדרישות/ שירותים שהמערכת מספקת . את רשימת הדרישות המלאה מצורפת בהמשך במסמך זה תחת סעיף "נספחים" .

** ניתן גם כן לעיין במפרט הדרישות תוכנה המלא SRS אשר נמצא במאגר הקוד שלנו במערכת ניהול הפרוייקט GITHUB בקטגוריה "WIKI" > SOFTWARE REQUIRMENT SPECIFICATION . במסמך זה תקבלו מענה לכל הקשור לדרישות ואת אופן מימושן.

5.2 הבעיה מבחינת הנדסת תוכנה/ אתגרים טכנולוגים

מדובר באפליקציה היברידית :

מה ולמה ?! כמעט בכל סיטואציה בה ארגון מעוניין לפתח אפליקציה למובייל עולה לדיון סוגיית ההתאמה למכשירים ולמערכות הפעלה. שימוש במתודולוגיית פיתוח נכונה ובחירה בטכנולוגיה מתאימה, מאפשרים להתאים כל אפליקציה לכל מערכת ההפעלה, בזמן קצר ובעלות נמוכה, מבלי לפתח מחדש חלקים גדולים של האפליקציה הגישה ההיברידית יותר ויותר ארגונים בוחרים לפתור את סוגיית ההתאמה למגוון המכשירים הניידים באמצעות פיתוח היברידי. בפיתוח ההיברידי מפרידים בין שכבת הקשר עם המכשיר לשכבת ממשק המשתמש (UI) והוא מציע מספר יתרונות שמעודדים להשתמש במנגנון זה .

איפה בדיוק האתגר ? מנגנון זה מסכן את האפשרות לספק את חוויית המשתמש הטובה ביותר והעשירה ביותר האפשרית מאחר והשימוש בגישה ההיברידית מאפשר לפתח את מרבית הממשק והלוגיקה (גוף האפליקציה) בצורה אחידה המשמשת את שתי מערכות ההפעלה. ההתאמות למערכות ההפעלה השונות מתבצעות בנפרד רק באזורים ספציפיים של האפליקציה תוך מניפולציה פשוטה של קבצי HTML ו- CSS. זה אומר שאין התמקדות במערכת הפעלה אחת ספיציפית על מנת להשיג חוויית משתמש טובה ביותר.

בנוסף לזה, הפיתוח ההיברידי מסכן את האפשרות לגישה מלאה לכל התקני החומרה של המכשיר. פיתוח היברידי לא נחשב למנגנון "הטוב ביותר" בגישה להתקי החומרה של המכשיר , לעומת פיתוח בשפה שהיא NATIVE. פיתוח ב-SWIFT למשל מספק גישה מלאה ומהירה להתקני החומרה של מכשירי אפל , אותו דבר לגבי ANDROID וסמסונג. הכלליות של הפיתוח ההיברידי מחלישה את יכולותיו בגישה להתקני חומרה .

יתר על זה, פיתוח היברידי לא בהכרח מספק את הגישה הכי מהירה ויעילה לנתונים בשרת של הארגון . אך זהו אתגר פחות מטריד כיוון שיש כל מיני מנגנוני גישה לזכרון שמאפשרים שליפה מהירה ויעילה של נתונים.

** איך אתגר זה בא לידי ביטוי בפרויקט שלנו ?

במימוש מערכת NFC לקריאת כרטיסים מקודדים .(ראו מסמך דרישות)

** יש לציין כי הדרישה לממש מערכת NFC הינה דרישת בחירה (אופנציונלית) ולא חובה !

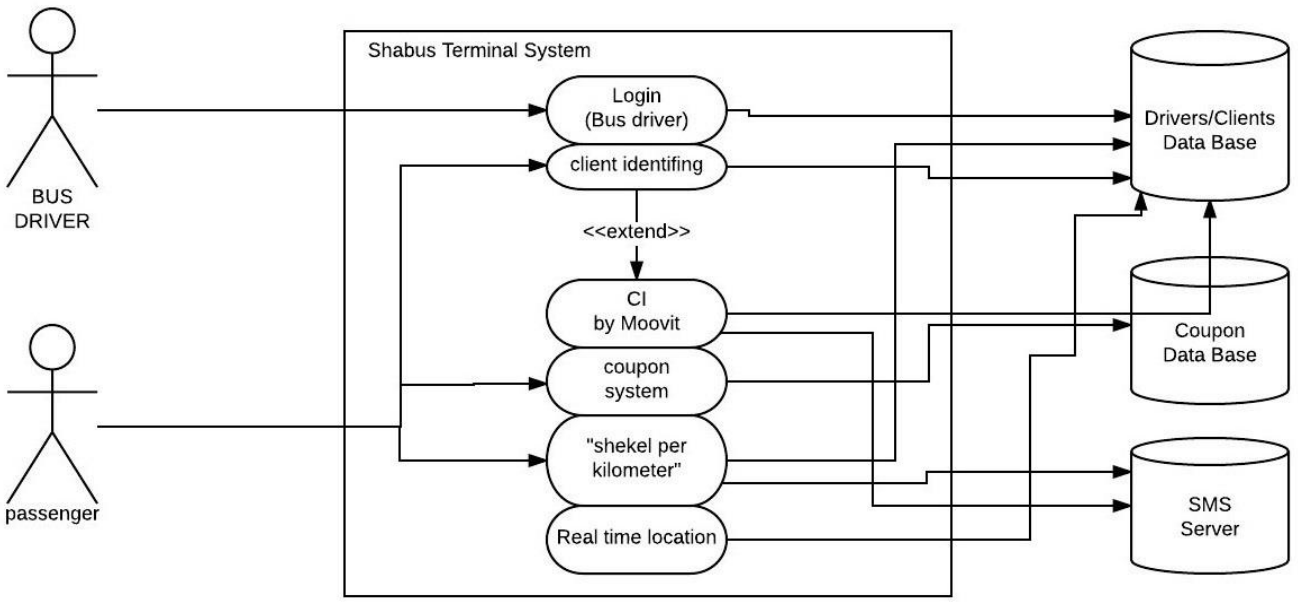
והצורך בה הולך ומתמעט וככל הנראה היא תודח מהרימה (לפי בקשתו של הלקוח).

אם הלקוח ירצה בעתיד להרחיב את הפרויקט , למשל הוספת מנגנון זיהוח ע"י טביעת אצבע , אז אתגר זה יכול להשפיע .

3. תיאור הפתון / כולל תיכון מפורט

1.3 ארכיטקטורה

לצורך הבנת "איך בנויה המערכת" , מצורף התרשים הבא :



תרשיים 1.3 : תרשים שימוש USE CASE

לפי התרשים לעיל , ניתן לראות שלמערכת יש שלושה חלקים (אזורים) עיקריים :

1. אלה שעושים שימוש בה .
2. שירותים של המערכת . שזה בעצם הדרישות של הלקוח , וזה למעשה משימות למימוש ע"י המפתחים. בהמשך הפיתוח , חלק זה
3. מסד נתוים .

הסברים :

אזור1 : משאבים (במקרה שלנו זה משאב אנושי) אשר עושה שימוש במערכת . ניתן לראות שאלה הם השחקנים בתרשים , הנהג והנוסע . בתחילת המשמרת הנהג עולה לאוטובוס , מפעיל את המערכת ומזדהה מולה באמצעות שם משתמש וסיסמא , ע"י שימוש (אנטראקציה) בשירות BUS DRIVER LOGIN.

הנוסע גם כן צריך להזדהות מול המערעת (בכל מקרה, בין אם הוא רשום באגודה או לא) באמצעות מספר טלפון . במידה והוא לא רשום באגודה , ניתנת לו הטבה חד פעמית לנסיעה חינם אך ורק אם הוא משתמש באפליקציית MOOVIT . אם הוא לא רשום והוא לא משתמש MOOVIT הוא לצערנו הרב יורד מהאוטובוס (עניין משפטי).

אזור 2 : זה בעצם הדרישות של הלקוח , וזה למעשה משימות למימוש ע"י המפתחים. בהמשך הפיתוח , חלק זה יחולק למשימות לביצוע (איטרציות) והוא יחולק גם כן לפי היבטים טכנולוגים שיתבררו בשלבים יותר מתקדמים.

אזור 3 : זהו השרת של החברה שבו מאוחסן כל המידע אודות כל בעלי העניין , נהגים , נוסעים , נסיעה, תחנות , מסלולים וכו.....
** ישנם שני כיוונים של גישה לאזור זה .בכיוון ימינה (בתרשים) שזה הוספת/עדכון מידע. ושמאלה שזה שליפת/קבלת מידע.

2.3 אבני יסוד

תת סעיף זה מציג את המאפיינים שמהווים אבן יסוד של המערכת אשר בלעדיהם המערכת לא תתקיים . והם נחשבים לנקודת ההתחלה של שלב הפיתוח .
המאפיינים הם זיהוי נהג (דף הכניסה למערכת) וזיהוי נוסע (דף בנפרד) . אלו שני מאפיינים בסיסיים שלא נייתן לוותר עליהם. הרי רוב שאר מאפייני המערכת נועדו לתת שירות והטבות לשני הגופים האלה (נהג/נוסע) . דבר זה הוא בלתי אפשרי אם לא נצליח לזהות אותם !

ובכך , הפיתוח של הפרוייקט מתחיל משני המאפיינים האלה . על מנת להשיג אבן יסוד בפיתוח שתהיה איזשהיא נקודת מוצא לצורך מימושים מתקדמים והרחבות.

**ניתן לעיין במסמכים הרליינטים במאגר הקוד שלנו כגון מפרט דרישות ומפרט תיכון לצורך קבלת מענה יותר עשיר ומפורט. (מצורף קישור בתחילת הדו"ח).

3.3 תיאור הפתרון המוצע

פיתוח אפליקציית מובייל שתנהל את נסיעת שבוס . מערכת WEB לא תשיג את הנדרש !
כיום ישנה מערכת WEB לצור רישום לאגודה אך העבודה כדף WEB מגבילה אפשאויות חדשות. לצורך פונקציונליות מורחבת אנא נצטרך לפתח אפליקציית מובייל.

4.3 כלים

היות ומדובר במערכת גדולה ומקיפה, אנו נשתמש במגוון רחב של טכנולוגיות וכלים.
קודם כל נזכיר כי אנו מפתחים אפליקציה היברידית , מסיבות שפורטו לעיל במסמך זה !
להלן רשימת הטכנולוגיות והלים שיסייעו לנו להגיע להישגים מרשימים ביותר ! :

- אנו נפתח את המערכת בשפת ANGULAR. שפה זו תמממש לכתיבת ה- CODE SOURCE
- נעשה שימוש ב IONIC או ב BOOTSRAP לצורך התאמת האפליקציה לגדלים וסוגים שונים של מכשירים . נשתמש במי שיתברר בהמשך כיותר טוב ויעיל.
- ANDROID STUDIO לצורך עטיפת המערכת לאפליקציית מובייל.
- התקשורת (שליפה/אחסון) מול השרת של שבוס נעשית ע"י שימוש ב REST API.
- נשתמש בשפות תכנות משניות כמו HTML ו JAVA SCRIPT לפי הצורך.
- נשתמש בשירותים חיצוניים (לרוב חינמיים) שיסייעו לנו בהשגת הנדרש . למשל GOOGLE MAPS , GPS , שרתי SMS וכו...
- נשתמש בשירות TWILIO לצורך שליחת הודעות SMS לפלאפונים של הנוסעים . יש לציין כי שירות זה אינו חינמי והוא כרוך בתשלום שנתי . (התקבלה הסכמת הלקוח) .
- נשתמש במערכת GUTHUB/DROPBOX לסנכרון וגיבוי הקוד.

** להסברים יותר מעמיקים ניתן לעיין במאגר הקוד תחת קטגוריית WIKI .

4. סקירת עבודות קודמות + סקר שוק

המעיין בפרוייקט שלנו ודרישותיו יבחין כי יש חלק מהשירותים שהוא מספק כבר קיימים בעבודות קודמות , וזה אכן כך ! אז למה בכל זאת ?!!

כידוע, יש לא מעט מערכות לניהול נסיעות מסוגים שונים והן מספקים שירותים מקיפים , והם חופפים את הפרויקט שלנו בחלק מהשירותים האלה ובמיוחד בכל הקשור לחישוב מסלולים ותחנות.

הדוגמה הכי טובה ומעוררת השראה לחקור היא אפליקציית MOOVIT . כידוע, אפליקציית MOOVIT היא אחת המובילות בתחום שלה : ניהול נסיעות. היא מנהלת נסיעות לכל מני סוגי תחבורה , בין אם זה תחבורה ציבורית (אגד/רכבת קלה) , תחבורה פרטית (דן וקווי לילה) אפילו מוניות שירות ומוניות של גיט טאקסי . והיא עושה זאת באופן הטוב ביותר ע"י סיפוק מגוון רחב של שירותים אודות הנסיעה , בעיקר לשימוש הנוסע !

דהיינו , אז למה צריך עוד מערכת ניהול נסעיות ובפרט שבוס ?!

1. הכלליות שולטת בעבודות קודמות : כמעט כל העבודות הקדומות נועדו לנהל נסיעה (כלשהיא) בכללי ולא לנהל נסיעה ספיציפית של אגף מסויים . הדבר שמעיד על חוסר התמקדות במאפייני נסיעה ספיציפית מה שמביא להפחתה במגוון ההטבות שניתן יהיה להציע בפני המשתמשים . למשל בנסיעה של שבוס יש אפשאות לעלות פעם אחת חינם ויש אפשאות לרכוש/לקבל קופונים להזלת מחיר הנסיעה . דבר זה מעיד על כך שנסיעת שבוס היא נסיעה מיוחדת בעלת תנאים מיוחדים , ולכן מן הסתם והראוי שהיא צריכה ניהול מיוחד על מנת להעניק למשתמשים הטבות משמעותיות.

2. פרספקטיבת שימוש ,כלומר למי נועדו לשמש העבודות הקודמות (סומן בצהוב למעלה) : רוב העבודות הקודמות (אם לא כולן) נועדו לשימוש הנוסעים וכמעט אין לנהגים שום אנטראקציה עם המערכת ! הדבר אומר שהמפתחים והמנדסים שבנו את המערכות האלה פיתחו אותן מפרספקטיבה של נוסעים. למשל ב MOOVIT , כל השירותים שהיא מספקת אודות הנסיעה מיועדים לשימוש הנוסע ולא הנהג , למשל זמני הגעת האוטובוסים לתחנה וחישוב מסלולים וחיפוש קווים , כל אלה נועדו לשימוש הנוסע . בשבוס לעומת זאת לנהג יש חלק עיקרי באפליקציה וצריך לזהות איזה נהג נמצא באיזה נסיעה.

3. מאפיינים חדשים שאינם קיימים בעבודות קודמות: כמו קילומטר בשקל , קופונים, זיהוי נהג/נוסע.(ראו רשימת דרישות בהמשך).

** בהינתן ההסבר הנ"ל , עולה הצורך באפליקציית ניהול ייחודית עבור שבוס. בטח שניעזר בשירותים שכבר קיימים ונתאים אותם לצרכים שלנו (כמו חישוב מסלולים וזמני הגעה) ובמידת הצורך נשפר אותם (כמו למשל דיווח מיקום אוטובוס שהוא לא מדויק). אך זה לא מכסה את המצופה ממערכת שבוס ,ולכן זה מייתר את פיתוחה .

5. נספחים

1.5 רשימת ספרות / קישורים

ארגון מחדש של התחבורה הציבורית בירושלים : מחקר של מבקר התחבורה הציבורית

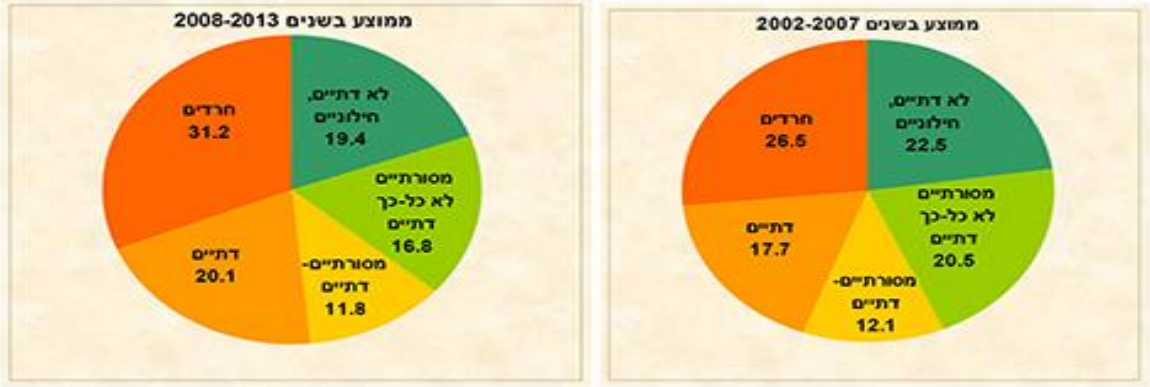
http://www.mevaker.gov.il/he/Reports/Report_266/8358fc21-447a-4f4e-8f54-a7e4274624b4/213-ver-4.pdf?AspxAutoDetectCookieSupport=1

סטטיסטיקות בירושלים : אחוז הדתיים לעומת החילונים . מידת שביעת רצון התושבים מתחב"צ

http://www.ynet.co.il/articles/0,7340,L-4523628,00.html

http://www.jerusalemnet.co.il/article/53932

תרשים 3. מידת הדתיות של יהודים בירושלים, 2013



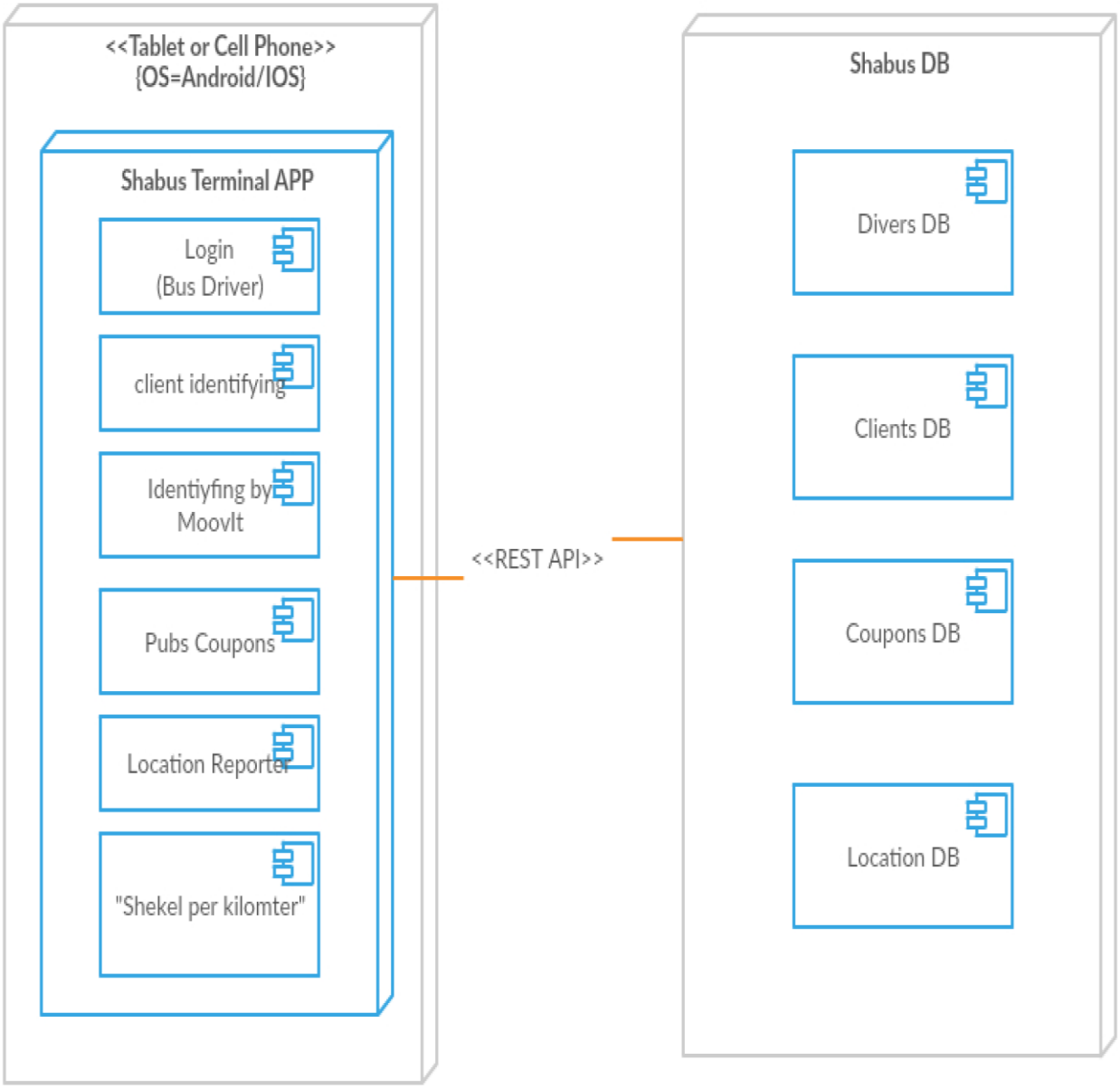
הספר דתיים בירושלים

http://jerusaleminstitute.org.il/.upload/datiim%20be%20yerushalaim.pdf

2.5 תרשימים נוספים / תיכון מפורט

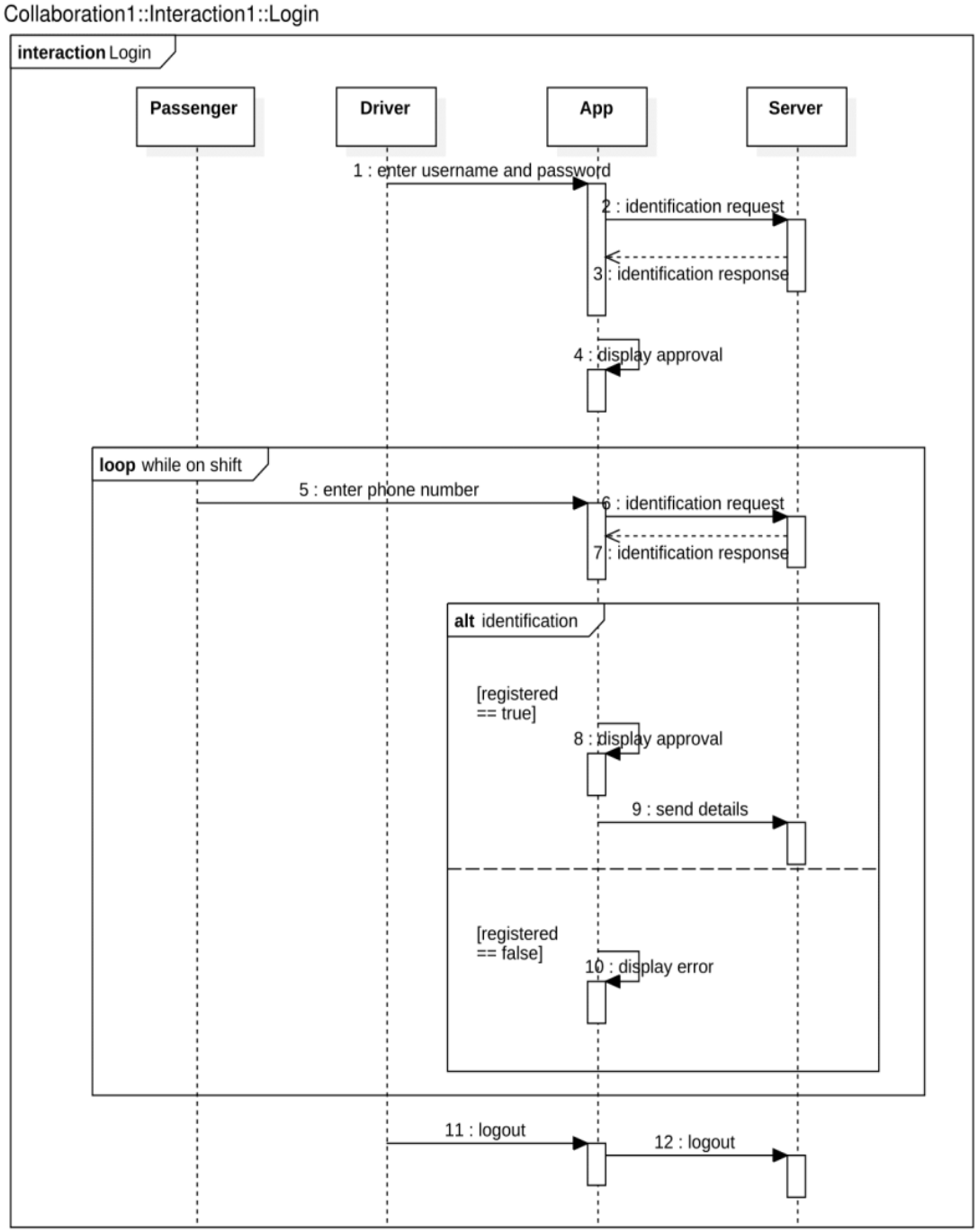
תרשימי תיכון :

1. תרשים הפצה DEPLOYMENT DIAGRAM

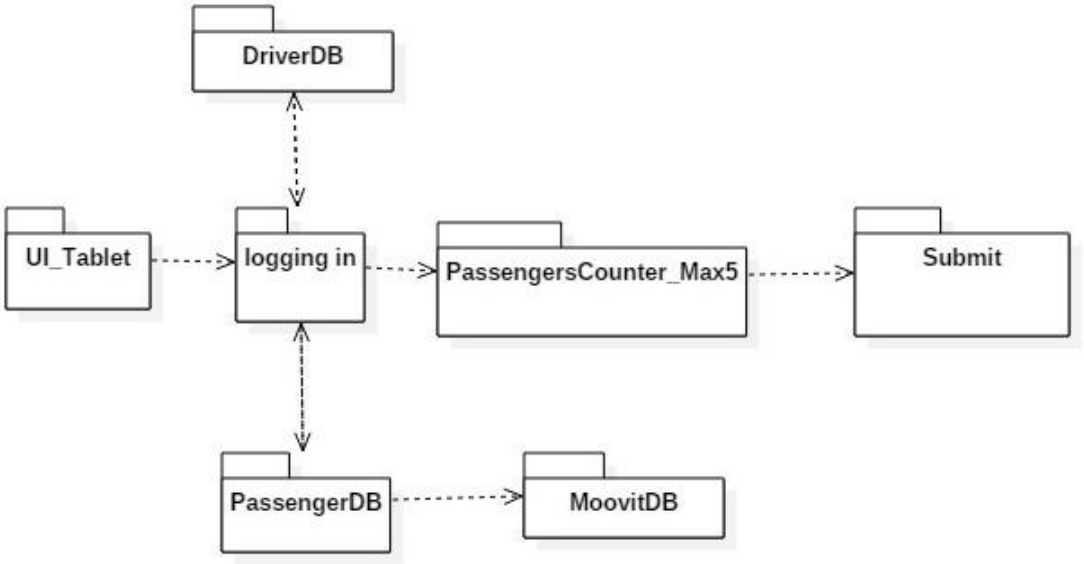


2. תרשים מחלקות CLASS DIAGRAM (UML)

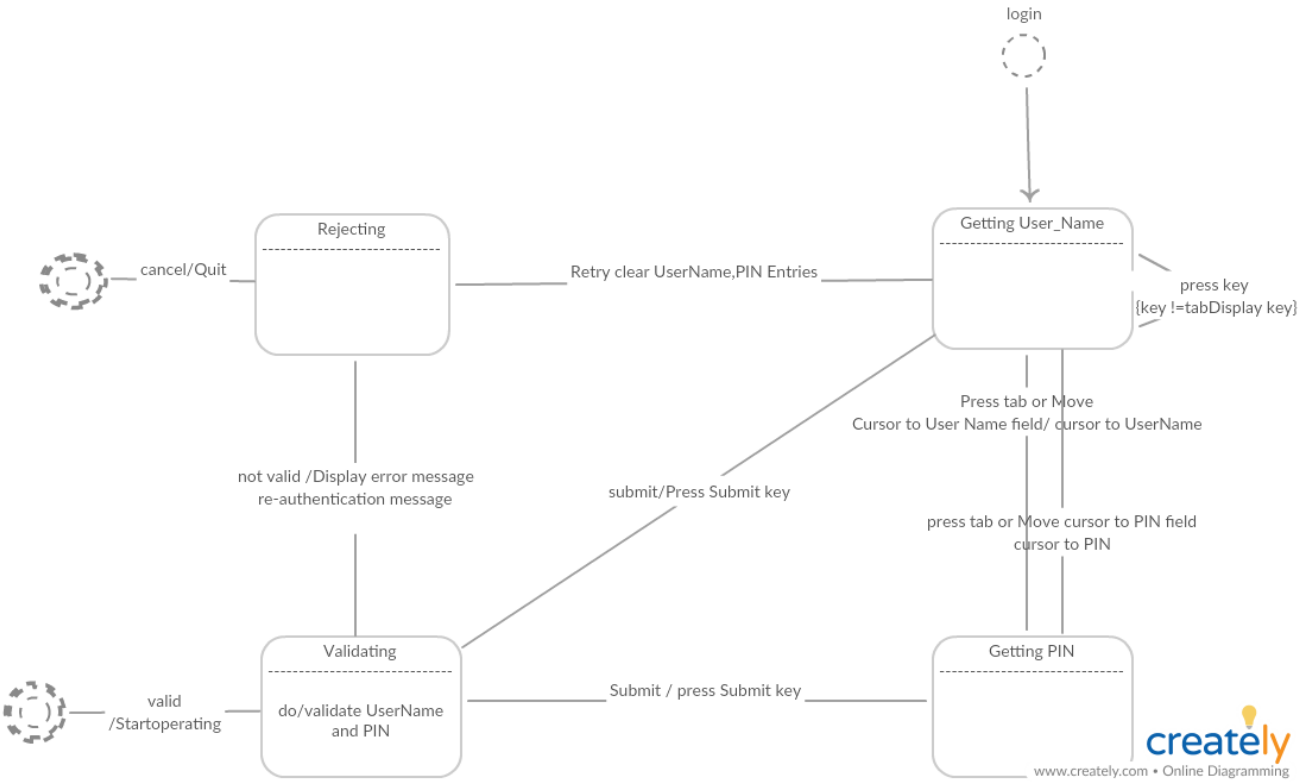
3. תרשים רצף SEQUNCE DIAGRAM



4. תרשים אופייקטים Object diagram



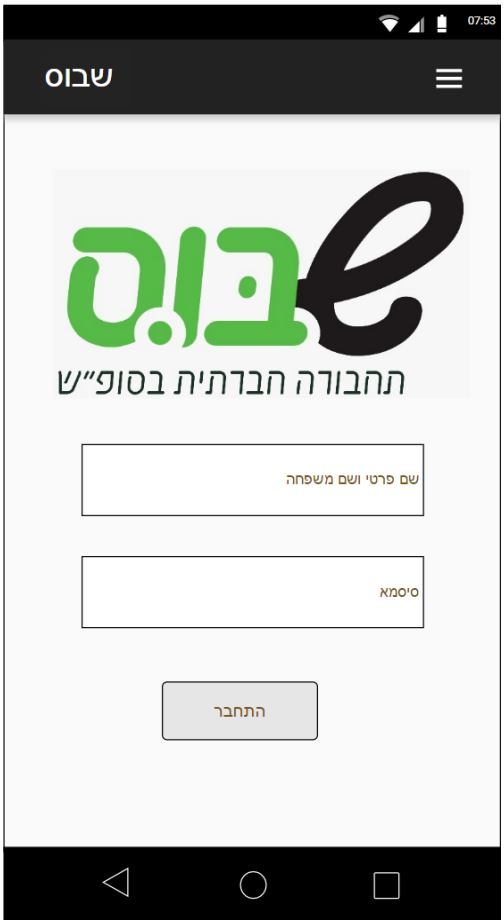
5. תרשים מצבים לתהליך זיהוי הנהג Driver Login State Chart



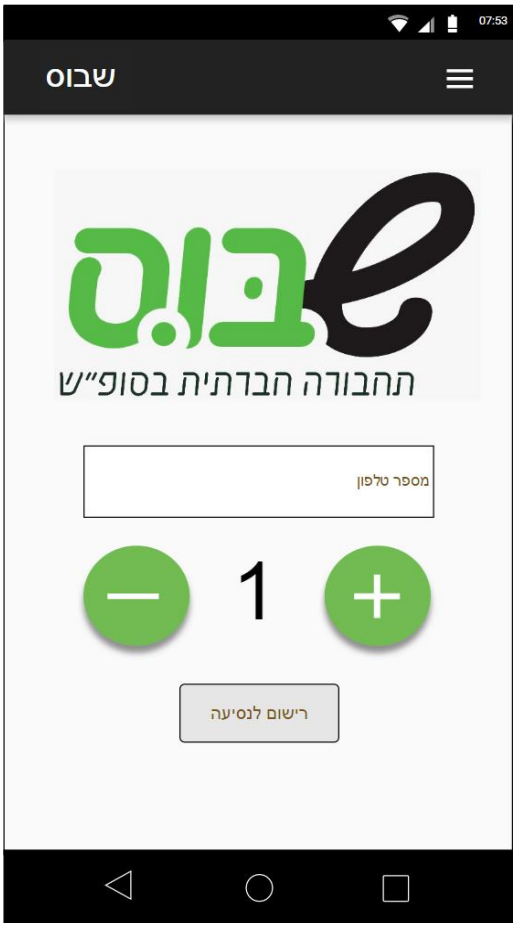
** תהליך זיהוי הנוסע מאוד דומה לתהליך זיהוי הנהג . ולכן התרשימי מצבים שלהם מאוד דומים.

מסכים צפויים מתוך USER MANUAL:

1. דף כניסה / זיהוי נהג :



2. זיהוי נוסע



** ניתן לראות מדריך שלם למשתמש USER MANUAL במאגר הקוד שלנו תחת קטגוריית WIKI בתוך מסמך תיכון דרישות SDS

3.5 תכנון הפרויקט (ראשוני)

תאריך	תוכן
25/10/2017	התכנסות פרויקטים עם האחראי האקדמי "אסף"
8/11/2017	פגישת התכנסות עם האחראי התעשייתי "נדב כהן"
20-25/1/2018	סיום פיתוח צד לקוח, עיצוב דפים
30/2/2018	אב טיפוס גרסת אלפא
15/4/2018	סיום פיתוח צד שרת
1/6/2018	גרסת פיתא (מסירה)

4.5 טבלת דרישות (URD)

מס	דרישה	צורך	תיאור
.1	זיהוי נהג	M	בתחילת המשמרת הנהג חייב להזדהות מול המערכת באמצעות שם משתמש וסיסמה .
.2	זיהוי נוסע	M	בעת העלייה לאוטובוס , הנוסע חייב להזדהות מול המערכת באמצעות מספר טלפון .
.3	דף MOOVIT	D	אם יתברר שהנוסע אינו רשום באגודה , אז ניתנת לו הטהב חד פעמית לנסיעה חינם אך ורק אם הוא משתמש באפליקציית MOOVIT .
.4	מערכת קופונים	D	שבוס מציעה עבור סוג מסויים של אנשים (ברמנים/סטודנטים) סוגים שונים של קופונים שמשמשים להוזלת מחיר הנסיעה.
.5	שקל לקילומטר	D	במסלולים מסויימים , שבוס מאפשרת מנגנון שקל לקילומטר שמחשב את מרחק הנסיעה עבור כל נוסע לפי מספר התחנות . והיא מחשבת את מחיר הנסיעה בהתאם עבור כל נוסע .
.6	דיווח בזמן אמת	D	המערכת צריכה לדווח את מקום האוטובוס מדי 30 שניות.
.7	מערכת NFC	O	שבוס שוקלים בעתיד לקודד כרטיס נסיעה מקודד בדיוק כמו הרב קו – לכן נצטרך מערכת שתקרא אותו.

הערות : M > MANDATORY > דרישת חובה
D > DESIRABLE > דרישה שכדאי שתהיה
O > ORTIONAL > לא קריטי – תלוי בזמן

** קישור לדף הדרישות המלא מטעם הארגון :|
https://github.com/murradkh/Shabus-Teminal-App/blob/master/materials/shabus.pdf

5.5 טבלת סיכונים

מס	סיכון	חומרה	תיאור
.1	מדובר באפליקצייה היברידית – אלמנט UI – חוויית משתמש נמוכה	8	באפליקציות אייפון סרגל הניווט ממוקם בתפריט (Bar) התחתון וכפתור ה- Back ממוקם בצידו השמאלי של החלק העליון. לעומת זאת, במכשירי האנדרואיד תפריט הניווט ממוקם בחלק העליון וכפתור ה- Back ממוקם על גבי מסגרת המכשיר ואינו מהווה חלק מהאפליקציה. לכן כאשר מעבירים ממשק משתמש שפותח עבור אייפון לאנדרואיד, מתקבלת אפליקציה שמרבית משתמשי האנדרואיד פשוט לא ידעו איך להשתמש בה. במקרה הגרוע, האפליקציה תגרום לחוסר שביעות רצון בקרב המשתמשים ולדעתם השלילית על הארגון שפיתח את האפליקציה. פתרון: עדכון שותף מול הלקוח.
.2	מדובר באפליקצייה היברידית – גישה לזכרון	6	אם יעלה מספר נוסעים בבת אחת (מתחנה אחת) אנו צריכים לזהות אותם במהירות המקסימלית כך שלא יווצר מצב שבו הנוסעים יחכו הרבה זמו בכניסה לאוטובוס . לכן אנחנו צריכים גישה מהירה ביותר לזכרון . ועצם העובדה שמדובר בפיתוח היברידי יכול לסכן זאת. פתרון : שימוש במנגנון גישה מהיר כמו REST API

.3	הערכות שגויות	5	ייתכן וההערעות שנתו עבור מימוש משימות כלשהן יתבררו כי שגויות וכי מימיוש משימה מסויימת לקח יותר זמן ממה שתיכננו. פתרון: אנו נבנה תכנון התחלתי וננסה ככל הניתן לעמוד בו ולשאוף לזמנים הרשומים בו .
.4	האפליקצייה יקרה משאבים	6	המערכת לצורך העניין צריכה לעדען נתונים אודות הנהג בעת הכניסה למערכת (כגון השם שלו ובאיזה שעה הוא התחבר ובאיזה תחנה הוא התחבר). אם נעשה זאת מדי יום אז שטח האחסון שלנו יילך וייקטן. פתרון: לשמור נתונים עדכניים ביותר . והשאר למחוק . למשל נשמור מידע רק מלפני שבוע ולא יותר.
.5	ציפיות גבוהות מהלקוח.	3	עדכון שוטף

6. תוכנית בדיקות

לפני שמסבירים חלק זה חשוב לציין שהדברים שנרשמים פה הינם דברים ממוקדים ומוצמצמים (רשום פה תכלס) . ** יש הפנייה מלאה לתוכנית הבדיקות בטבלה בתחילת מסמך זה ! יש שם גם את כל המקורות שבהם נעזרנו בבחירת תוכנית בדיקות מתאימה ואופן ביצוע בדיקות אלה !

1.6 בדיקות יחידה . מה זה ?

כידוע , ישנן סוגים שונים של בדיקות תוכנה, אנו נתמקד בבדיקות יחידה .

מהי בדיקת יחידה ? רמה בתהליך בדיקת התוכנה שבה יחידות/רכיבים בודדים נבדקים.

המטרה היא לוודא שכל יחידה/רכיב במערכת מתפקד כפי שתוכנן.

מהי יחידה ? יחידה היא החלק הבדיק הקטן ביותר במערכת כלשהיא. ליחידה יש בדרך כלל מספר קלטים ופלט יחיד . בתכנות פרוצדואלי , יחידה יכולה להיות תוכנית בודדת , פונקציה , תהליך וכו בתכנות מונחה עצמים , היחידה הקטנה ביותר הינה פונקציה .

בדיקות יחידה מבוסס על " בדיקת קופסא לבנה " ; זהו סוג בדיקות המבוסס על ניתוח המבנה הפנימי של היחידה/רכיב.

2.6 למה בדיקות יחידה ?

1. בדיקות יחידה מעלה את הבטחון ל שינוי/עדכון/ תחזוקת הקוד. בדיקות יחידה , אם מיושמות טוב ובמידה הן מופעלות מדי פעם שמתרחש שינוי בקוד , אנו יכולים מיד ובקלות לזהות ולאתר פגמים שנוצרו כתוצאה מהשינוי הזה. בנוסף , אם הקוד מההתחלה היה כתוב בצורה שלא מאפשרת לממש בדיקות יחידה , ההשפעה של השינוי הינה כמעט אפסית.

2. סיכוי יותר גדול לשימוש חוזר בקוד: על מנת לאפשר מימוש בדיקות יחידה , הקוד צריך להיות מודולרי. הדבר שעושה את הקוד קל יותר לשימוש חוזר.

3.פיתוח מהיר יותר של הקוד :איך ? אם באזור כלשהוא בקוד לא מיושם בדיקות יחידה . מה עושים בדרך כלל ? בודקים אותו ידנית ע"י ניסוי וטעייה , מכניסים קלטים כלשהם ומקווים שיתנו את הפלט הרצוי. דבר זה גוזל זמן רב מהזמן הכולל של הפיתוח. לעומת זאת , אם מיושמים בדיקות תוכנה , הבדיקות נכתבות פעם אחת . אומנם שכתיבת הבדיקות לוקחת זמן, אבל זה משתלם בזמן שלוקח לביצוע הבדיקה . חוץ מזה שבדיקות יחידה הן הרבה יותר איכותיות מניסוי וטעייה.

4. המחיר לתחקון פגמים שמגלים באמצעות בדיקות יחידה הינו נמוך יותר לעומת פגמים שמגלים ברמות/שכבות גבוהות יותר. מחיר יכול להיות זמן , מאמץ וכו'...

5.תהליך ה- Debugging הוא הרבה יותר קל : כשבדיקה כלשהיא נכשלת , צריך לדבג רק את העדכון האחרון של הקוד . לעומת זאת , בבדיקות מסוגים אחרים , צריך לדבג שינויים בקוד מתקופות יותר ארוכות .(שינויים לפני יום ,שבוע, חודש...).

סיבה נוספת חשובה ביותר :

נניח שיש לנו תוכנה שמורכבת משני רכיבים/יחידות . והבדיקה היחידה שאנו מבצעים היא בדיקת מערכת (אנו מדלגים על בדיקות יחידה/אנטגרציה) . במהלך הבדיקה אנו מגלים BUG , איך נדע מה גרם לזה ? :

- האם הוא נוצר בעקבות שגיאה שהתרחשה ביחידה מס 1?
- האם הוא נוצר בעקבות שגיאה שהתרחשה ביחידה מס 2?
- האם הוא נוצר בעקבות שגיאה שהתרחשה ביחידה מס 1 ו-2 ?
- האם הוא נוצר בעקבות שגיאה שהתרחשה בממשק שמשלב את שתי היחידות ?
- האם הוא נוצר בעקבות שגיאה בבדיקה עצמה ?

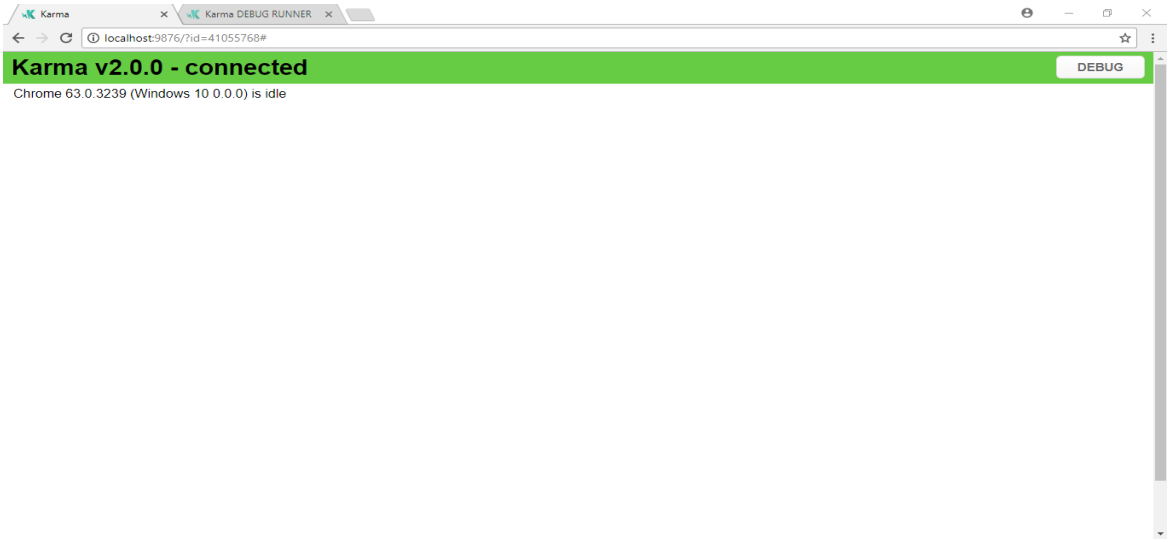
רואים בבירור שאנו נמצאים בבעיה גדולה בלי התייחסות לבדיקות יחידה בשלב כלשהוא בפיתוח.

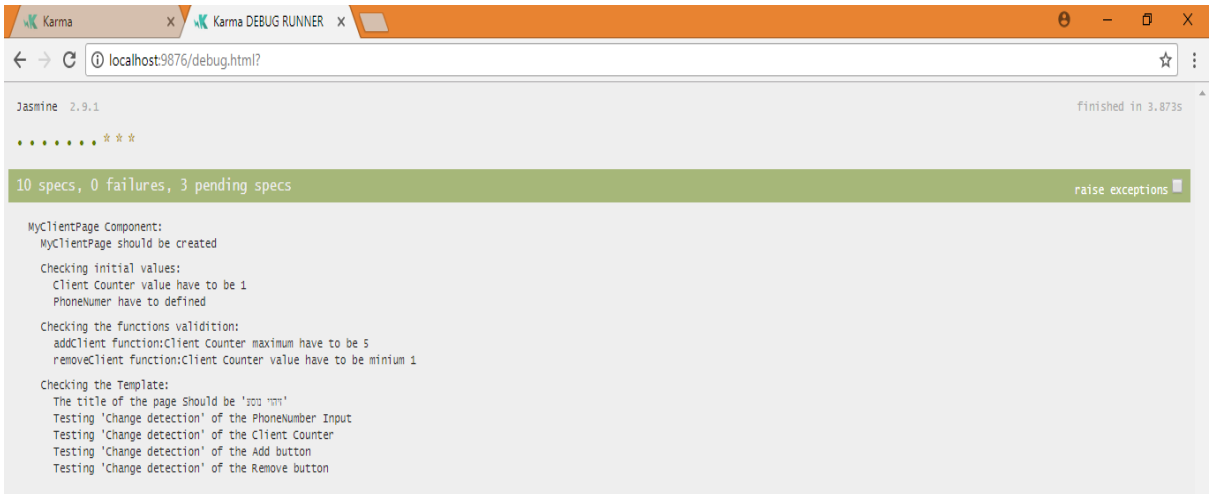
3.6 אופן ביצוע בדיקות יחידה

בקישור בהא מוסבר בפירוט אופן ביצוע בדיקות יחידה ב- ionic
http://getliner.com/Oo33J

** ניתן לראות את המימוש שלנו בפועל של בדיקות יחידה בקוד שלנו שמפורסם ב GITHUB שלנו (מצורף קישור בטבלה בתחילת מסמך זה) . שם ניתן לעיין בקבצים מסוג (עם סיומת) spec שבהם נכתבו הבדיקות.

הרצת הבדיקות נעשית ע"י הממשק Karma ו- Jasmin framework שנמצאים בקישור לעיל.





7. מסקנות ודברים חשובים להמשך

מסקנות :

- בחירת מנגנון בדיקה עבור פרוייקט תוכנה כלשהוא אינה בחירה חד משמעית , כלומר כשבוחרים להשתמש בתכנית בדיקות מסויימת זה ניתן לשיקול דעתם של המפתחים . במילים אחרות נדרש לפי שיקול דעתם של האחראים על הפרוייקט לעשות Trade Off בין הסוגים השונים של בדיקות תוכנה , ה Trade Off הזה נקבע לפי כל מיני קריטריונים , בין היתר מהי המטרה שבשבילה נועד הפרוייקט , ואיזה רכיבים עיקריים יש בפרוייקט שאי אפשר לוותר עליהם, הכוונה שאי אפשר לוותר עליהם היא שאי אפשר יהיה להשיק את המערכת מבלי שרכיבים אלה יעברו בדיקה רצינית . כידוע שישנם רכיבים פחות מרכזיים בכל פרוייקו שהוא שאפשר להסתפק בבדיקה אחת מקיפה לגביהם ואין כל כך צורך בבדיקה ממוקדת לכל אחד ואחד.

ע"י זה שאנחנו מגלים את אופן פעולת/מטרת הפרוייקט ורכיבים מהסוג הנ"ל זה יעזור לנו לקבוע באיזה מידה אנו מעדיפים להשתמש בסוג מסויים של בדיקה על פני השני.

יש לציין כי ברוב המקרים שואפים האחראים על הפרוייקט לשלב בין הסוגים השונים של בדיקות לפי הצורך.

** בפרוייקט שלנו מוסבר בדוח הזה לעיל (סעיף 6) איזה סוג נבחר ולמה.

חשוב מאוד לזהות בשלבים מוקדמים רכיבים/חלקים קריטיים על מנת לסייע בשלבים קצת יותר מתקדמים בבדיקת המערכת.

דהיינו , זיהוי ומיון טוב של רכיבים אלה נעשה ע"י תיכון מפורט ומקיף של המערכת. הדבר שמעיד על חשיבותה הרבה של העבודה הפורמלית "ה – לא מעשית" .

**** האגדה מספרת שמתכנתים (במיוחד סטודנטים) פחות מתחברים לחלק הפורמלי הזה ולרבות לא משקיעים בו (כנראה שהיא צודקת) . מאחר וחווינו על בשרנו את חשיבתו הרבה של חלק זה אנו יודעים להעריכו טוב מאוד, בינתיים ובעבודות עתידיות.**

הערות להמשך הדרך :

בהמשך הפיתוח , אנו שוקלים לממש/לנסות עוד מנגנון של בדיקות שזה בדיקות "קצה-לקצה" E2E -ENDTO END . השיקול הזה עולה מאחר ובשלב כלשהוא צריך לבדוק את המערכת **כולה** . כלומר איך היא מתפקדת **כיחידה אחת** , ולא איך כל יחידה פועלת בנפרד. וזה בדיוק מה שמספק מנגנון זה. דחיית השימוש במנגנון זה לשלבים מתקדמים בפרוייקט מתבקשת מאחר ואופן פעילות מנגנון זה אינה מתאימה/תורמת למערכות שהם עדיין בשלבי בנייה. הרי מנגנון זה בודק את המערכת כולה , איך נעשה זאת אם היא עדיין לא קיימת בשלימותה.

** הערות חשובות ביותר :

- הרבה דברים שצריך לפרט בדוח אלפא פורטו כבר בדוח ההצעה . לכן הרבה פרקים לא השתנו בדוח זה. בשלב הזה (אלפא) התיכון צריך להיות מפורט , אבל אצלנו כבר בדוח הקודם פירטנו את התיכון. התווספו דברים פה ושם . את הדברים החדשים סימנתי בצבע תכלת.

נא לשים לב להפניות שקשורות לתוכנית הבדיקות. יש שם חומר משמעותי שלא מתאים/אפשר לרשום בדוח.

נסיעה טובה !