

# המחלקה להנדסת תוכנה

## פרויקט גמר – תשע"ו

### מערכת לניהול משאבי הארגון

mobile ERP system

מאת

מיתר שוקרון

זאב מלומיאן

מנחה אקדמי: מר שי תבור  
אחראי תעשייתי: מר עופר פישלוביץ'  
רכז הפרויקטים: פרופ' אסף שפינר  
תאריך: אישור: תאריך: אישור: תאריך: אישור:

מערכות ניהול הפרויקט: בשלב ההצעה יש להוסיף זאת בטופס התנעה

#	מערכת	מיקום
1	מאגר קוד	<a href="https://github.com/meitarsh/m.s-aluminium-manager-app">https://github.com/meitarsh/m.s-aluminium-manager-app</a>
2	יומן	<a href="https://calendar.google.com/calendar/embed?src=t1f2ojv5arrqonei6h09i5fld0%40group.calendar.google.com&amp;ctz=Europe/Athens">https://calendar.google.com/calendar/embed?src=t1f2ojv5arrqonei6h09i5fld0%40group.calendar.google.com&amp;ctz=Europe/Athens</a>
3	ניהול פרויקט (אם בשימוש)	
4	הפצה	<a href="https://github.com/meitarsh/m.s-aluminium-manager-app/releases">https://github.com/meitarsh/m.s-aluminium-manager-app/releases</a>

## תוכן עניינים

3.....	מילון מונחים, סימנים וקיצורים.....	
4.....	מבוא.....	1.
4.....	תיאור הבעיה.....	2.
4.....	דרישות ואפיון הבעיה.....	
4.....	הבעיה מבחינת הנדסת תוכנה.....	
5.....	תיאור הפתרון.....	3.
5.....	מהי המערכת.....	
5.....	מצבי עבודה ותכנון הפתרון.....	
6.....	תיאור הכלים המשמשים לפתרון.....	
7.....	ריכוז פרויקטים דומים.....	4.
9.....	נספחים.....	5.
9.....	רשימת ספרות \ ביבליוגרפיה.....	א.
10.....	תרשימים וטבלאות.....	ב.
18.....	תכנון הפרויקט.....	ג.
19.....	טבלת סיכונים.....	ד.
19.....	רשימת/טבלת דרישות.....	ה.

- מילון מונחים, סימנים וקיצורים

**הלקוח** – החברה מ.ש. אלומיניום, חברה להפקת מוצרי אלומיניום, אשר מוצר הפרויקט מיועד לשימוש שלה.

**השרת** – מחשב אשר נמצא בבניין הלקוח אשר מריץ מחובר אל האינטרנט בשאיפה כל הזמן ותפקיד המחשב הזה הוא להחזיק בנתוני החברה ולספק שירותי שליפת וכתובת נתונים אונליין, לספק שירותי מייל ושירותים אחרים שיתכן והלקוח מעוניין בהם.

**המסד נתונים** – מערכת האחסון בשרת שבנה נשמרים הנתונים ארגוניים של הלקוח כולל משתמשים מורשים, פרויקטים שונים שהלקוח מבצע ומצבם, עובדים ושכריהם וכו'.

**המוצר/אפליקציה** – התוצר הסופי של הפרויקט, יהיה אפליקציית אנדרואיד שמרצה את דרישות הלקוח ודרישות קהל היעד (מנהלי העבודה של הלקוח) גם מבחינת פונקצנאליות וגם מבחינת ממשקיות וחווית משתמש.

**המכשיר** - מכשיר סלולארי העובד על מערכת הפעלה אנדרואיד, יהיה בעצם הרכיב חומרה שעליו תרוץ המערכת שלנו.

**מצב offline** - מצב בו המכשיר לא מחובר על השרת האינטרנט (מצב לא מקוון) ומסיבה זאת אין חיבור אל השרת, במצב זה כל הפקודות שנרצה לשלוח אל השרת כדי לשנות את התוכן של השרת ישמרו על המכשיר וישלחו שוב במועד אחר כאשר אנו יודעים כי המכשיר עובד במצב מקוון.

**SQLite** - מערכת לניהול מסדי נתונים שיש לה תמיכה טבעית במערכת הפעלה אנדרואיד, אנו נשתמש בה בתור מנהלת תוכן הקבצים המקומיים ששמורים על המכשיר כזכרון cache בין השרת למוצר.

**Windows Server 2012 R2** - מערכת ההפעלה (או התוכנה) שמופעלת כל הזמן על השרת, היא מספקת לנו שירותים דרך האינטרנט שמספקים את השרת ואת הלקוח, ביניהם Microsoft SQL Server ויכולת הזדהות.

**Microsoft SQL Server** - מערכת לניהול מסדי נתונים בצד השרת, אחראית על לשמור מידע על השרת ולשלוח מידע החוצה ממנו ולספק רמת אבטחה מינימלית, המערכת תספק אותנו כשירות דרך האינטרנט.

## 1. מבוא

המערכת שלנו באה לתת מענה לציבור מנהלי העבודה של הלקוח שלנו שמוצאים את תהליך הדיווח של מצבי הפרויקט שלהם ועדכונים אחרים מאתרי הפרויקט אל השרת כתהליך ארוך ומייגע, תלוי מצב תקשורת באתר הלקוח ודורש ידע טכנולוגי רב.

המערכת שלנו תיתן מענה ישיר לקהל היעד שלה ותקל על תהליך התקשורת בין המכשיר הסלולארי לבין השרת כך שתהליך הדיווח יהיה לא יותר מאשר שליפת סמארטפון והקלדת נתונים, המערכת תדאג להסתייר מפני המשתמש את התקלות שהוא יקבל עקב כחוסר בתקשורת ככל שניתן ותדאג לשלוח את הנתונים בכוחות עצמה או לפחות ללא התערבות רבה מצד המשתמש.

ממשק המערכת יהווה פיצ'ר מפתח במערכת שלנו כיוון שבמהלך תוכנית העבודה וגם לקראת הסיום שלה ממשק המשתמש ישתנה ויתחזק במקביל לדרישות קהל המשתמשים ומודולריות התוכנה תהווה יכולת שינוי והוספת פיצ'רים פשוטה מצד הלקוח ככל שעולה הצורך במהלך חיי המוצר.

## 2. תיאור הבעיה

### דרישות ואפיון הבעיה

מצב הנוכחי של הלקוח שלנו כיום, הוא שקיים אצלו שרת במבנה הארגון ע"ג מערכת ההפעלה

Microsoft Windows Server 2012

אצל הלקוח יש כעשרה עובדים בתפקיד מנהלי פרויקטים אשר נדרשים לדווח את מצב הפרויקט באופן שוטף, אבל במסגרת הטכנולוגיה שקיימת אצלם כיום הם מזינים דו"ח אחת לחודש אשר מפרט את התקדמות הפרויקט והסטטוס שלו.

לעיתים קרובות מנהלי הפרויקטים נמצאים באזורים ללא קליטה ובכך אין להם גישה לאינטרנט, ולכן אין להם גישה ישירה לשרת ויכולת דיווח מידית.

הלקוח רוצה פתרון טכנולוגי אשר יאפשר למנהלי עבודה לדווח בזמן אמת ממוקם הפרויקט מהמכשיר הסלולארי שלו את מצב הפרויקט לשרת מבלי להיות תלוי בתקשורת שלו (האם קיים קליטה או לא) ובכך ללא צורך לזכור או לרשום בצד את הדוחות אלא לדווח במיקום ו"לשכוח מהעניין" – כלומר להקל על תהליך הדיווח מצד מנהל העבודה.

הלקוח שלנו מוכן לקבל עדכונים אחת לשעה (או כאשר למנהל עבודה יש תקשורת מקוונת לאינטרנט) ואצל כל מנהלי העבודה קיימים מכשירי אנדרואיד והלקוח מוכן לפיתוח למערכת אנדרואיד.

### הבעיה מבחינת הנדסת תוכנה

הבעיה הראשונה שאנו צריכים ללמוד להתמודד איתה זה דרכי התקשורת והזדהות מול שרת הלקוח, במצב הנוכחי שרת הלקוח מאובטח ודורש הזדהות לפני שתהליך ה session מתחיל, לכן אנו צריכים ללמוד על API ועל תהליך ההזדהות ברמת התוכנה כדי שנוכל להטמיע אותו ולדאוג לאבטחה מקסימלית בצד המשתמש, לאחר מכן אנו צריכים לדעת איך לתקשר עם רכיבי ה SQL של הלקוח, זה כנראה יתבצע באמצעות jdbc

הבעיה השניה שנצטרך להתמודד איתה זה ההטמעה של התקשורת במצב offline - נצטרך לבנות מסד נתונים נוסף על גבי המכשיר שיחזיק נתונים מקומיים וגם ישמור עדכונים לשרת במכשיר עד

שהמכשיר יחזור למצב מקוון, וכמובן שצריך להטמיע מעין service שידע לשלוח מחדש ולזהות חיבור מחדש לאינטרנט כדי לשלוח את הנתונים שוב. הבעיה השלישית שנצטרך להתמודד איתה זה זיהוי בעיות אבטחה באפליקציה שלנו, אנו מניחים כי מכשיר המשתמש לא יגיע לידיים הלא נכונות ובגלל שללא גישת root אין גישה לזכרון האפליקציה שלנו במכשיר (יש מנגנון שכאילו פותח user חדש לכל אפליקציה וזכרון האפליקציה הוא תיקיה שרק ל"משתמש" הזה יש גישה אליו), אנו צריכים לבדוק ולאבחן את שקיפות החבילות שיוצאות מהמכשיר ולנסות ככל שניתן למנוע זליגת מידע לא רצוי.

### 3. תיאור הפתרון

#### מהי המערכת

המערכת תיתן למשתמש את הפונקציות הבאות

1. הזדהות מול השרת
2. שאיבת רשימת הפרויקטים שהמשתמש מונה עליהם
3. שאיבת נתונים על פרויקטים שהמשתמש בוחר להסתכל עליהם
4. עדכון נתונים שהמשתמש בוחר לעדכן בשדות ייעודיים (ישנם שדות שלא ניתן לעדכן)
5. תצוגה גרפית של נתונים על מצב הפרויקט בהתאם לדרישות הלקוח

#### מצבי עבודה ותכנון הפתרון

המערכת תכלול שני מצבי עבודה עיקריים:

- מצב עבודה אונליין – בה יש תקשורת וסנכרון עם שרת הלקוח, במצב זה נשמרים ומתעדכנים הנתונים ע"ג המכשיר וכל פעולה שהמשתמש עושה ע"ג שרת הלקוח נשלחת על שרת הלקוח
- מצב עבודה offline - במצב זה אין תקשורת וסנכרון עם שרת הלקוח, מוצגים נתונים שעודכנו לאחרונה ע"ג המכשיר וכל פעולה ע"ג שרת הלקוח נשמרת במכשיר ותשלח רק אחרי שנחזור למצב עבודה online

הפתרון יהיה מערכת שמשתמשת ברכיבי Microsoft כדי להזדהות מול השרת, שימוש ב driver של jdbc כדי לבצע פעולות שאיבת נתונים מהשרת או עדכון השרת, ושימוש ב sqlite כדי לשמור טבלאות ונתונים ע"ג המכשיר ולקרוא אותם לפי הצורך

בגלל שהתוכנה עובדת מול תוכנת שרת קיימת, אנו נצטרך למצוא סביבת עבודה מקבילה או להשתמש בתוכנת הלקוח בתור סביבת עבודה במהלך הפיתוח, בגלל שקיים סיכון למידע ששמור ע"ג השרת אם אנו עובדים על גבי השרת, ננסה להשתמש במכונה ווירטואלית כדי לעשות סביבה משלנו קרובה ככל שניתן לסביבה של הלקוח, או לבקש מהלקוח "נישה" בחלק השרת שתהיה בטוחה למניפולציות

אופציה שלישית היא להכריח את הלקוח לבצע גיבויים תקופים, אופציה שננסה להתנער ממנה עקב אי הנוחות שיגרר כתוצאה מאותה האופציה

#### תיאור הכלים המשמשים לפתרון

בפרויקט נשתמש במסגרת הפיתוח של Android Studio כדי לפתח את האפליקציה שלנו, את סביבת הפיתוח אנו נריץ בשני מערכות הפעלה שונות ( Windows ו Manjaro Linux ), שפת התכנות העיקרית איתה נעבוד תהיה Java שאליה יצורפו כל הכלים שנכללים בספריות הפיתוח של Android

הפרויקט ינוהל לחלוטין ע"י מערכת GitHub שתנהל לנו את

1. מאגר הקוד
2. יומן אירועים
3. לוח מטלות

#### 4. ריכוז פרויקטים דומים

קיימים מספר פרויקטים שיש להם רמת דמיון לפרויקט שלנו ביניהם

##### WhatsApp

אפליקציית WhatsApp מספקת שירות הודעות בין משתמשים שונים דרך שרת WhatsApp שמספק את השירות הזה דרך האינטרנט, הודעות WhatsApp שנשלחו לשרת והתקבלו בשרת (או אפילו אצל הנמען) בהצלחה מקבלות אינדקציה באמצעות סימני וי



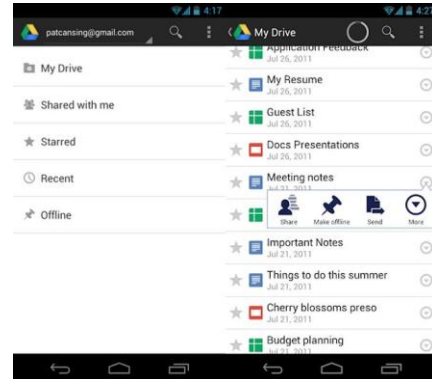
והודעות שלא הצליחו להתקבל בשרת מקבלות סימן של שעון, ונשמרות באפליקציה (בזכרון cache מקומי) עד כאשר ישנו חיבור חוזר אל האינטרנט ואז נשלחות



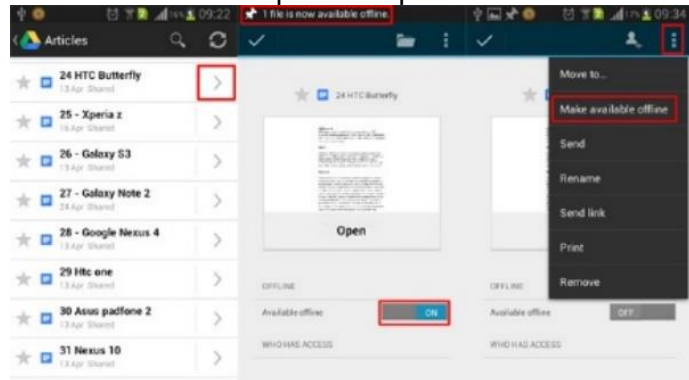
באותה רמה – אנו רוצים לממש מנגנון זה אבל לשרתי Microsoft SQL Server 2012, שימומש בצורה דומה לצורה שבה הוא ממומש באפליקציית WhatsApp

##### Google Drive

אפליקציית Google Drive מספקת שירותי ענן לאחסון קבצים



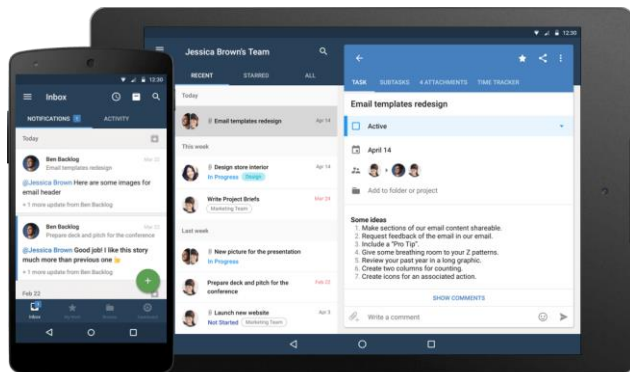
מעבר לסיפוק שרות זה, Google Drive נותנת למשתמש יכולת לשמור קבצים מסוימים בזיכרון המכשיר שהמכשיר יכול לקרוא את הקבצים האלו ללא כל גישה אל האינטרנט



באופן דומה אנו רוצים שהאפליקציה שלנו תציג רשימת פרויקטים ומידע נוסף שנשמר על זיכרון המכשיר על הפרויקט ללא כל גישה אל האינטרנט

### Wrike

אפליקציית Wrike מספקת שירות לניהול פרויקטים במאגר כלשהו ברשת, אפליקציית Wrike נותנת שירות לשמירת לו"ז, מיון לפי נושאים, ניהול משימות ותתי משימות ותצוגה של חיי הפרויקט בצורה גראפית



באופן דומה אנחנו רוצים לספק שירותים דומים ברעיון אבל לא זהים במהותם, כלי עזר למנהלי הפרויקטים של הלקוח שלנו, שיספקו את השירותים שהם צריכים במדויק לסיוע לניהול הפרויקט במהלך הפרויקט נרצה להטמיע שירות זהה למסופק ב Wrike שמציג את חיי הפרויקט בצורה גראפית, אבל בצורה יותר מפורטת ומותאמת לפרויקטים של הלקוח שלנו.



5. נספחים

א. רשימת ספרות \ ביבליוגרפיה

1.

Professional Microsoft SQL Server 2012 Integration Services by Brian Knight, Erik Veerman, Jessica M. Moss, Mike Davis, Chris Rock, 14 March 2012

[https://books.google.co.il/books?id=7\\_g0SocQ3CcC&sitesec=buy&hl=iw&source=gbs\\_atb](https://books.google.co.il/books?id=7_g0SocQ3CcC&sitesec=buy&hl=iw&source=gbs_atb)

2.

Database Programming with JDBC and Java by George Reese, 2000

[https://books.google.co.il/books?id=oPbGi0I0ZHEC&sitesec=buy&hl=iw&source=gbs\\_atb](https://books.google.co.il/books?id=oPbGi0I0ZHEC&sitesec=buy&hl=iw&source=gbs_atb)

3.

The Android Developer's Collection (Collection) by James Steele, Nelson To, Shane Conder, 9 December 2011

[https://books.google.co.il/books?id=3Wi2gwGoZZ0C&sitesec=buy&hl=iw&source=gbs\\_atb](https://books.google.co.il/books?id=3Wi2gwGoZZ0C&sitesec=buy&hl=iw&source=gbs_atb)

4.

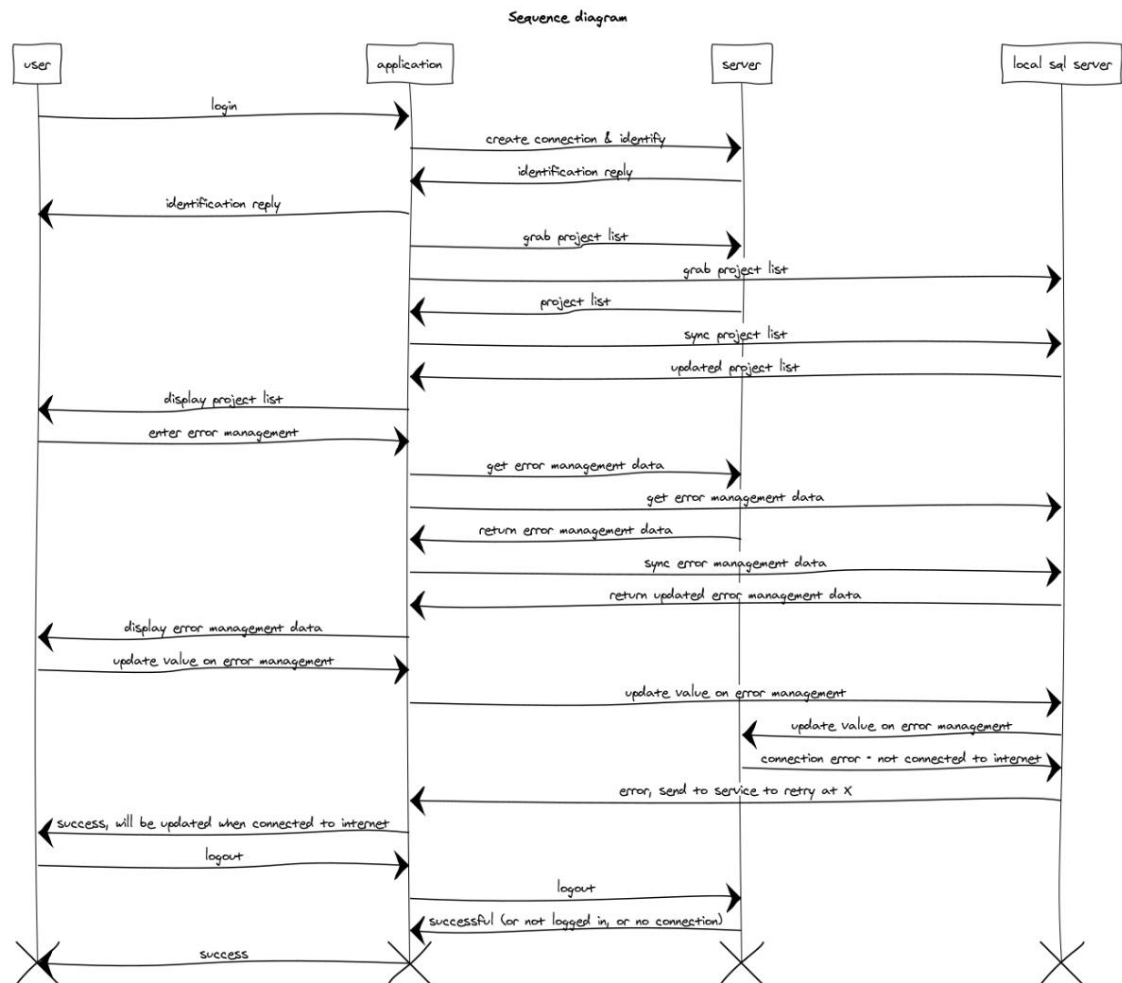
Getting Started with JDBC By David Reilly, 5 June 2006

<http://www.javacoffeebreak.com/articles/jdbc/>

ב. תרשימים וטבלאות

**דיאגרמת מצבים – מצב offline פתע**

על מנת שנוכל להסביר את חשיבות המימוש של מצב offline אנו מתארים  
בנספח זה את מקרה בודד מבין המצבים האפשריים כאשר שירות האינטרנט  
או החיבור אל השרת נופלים

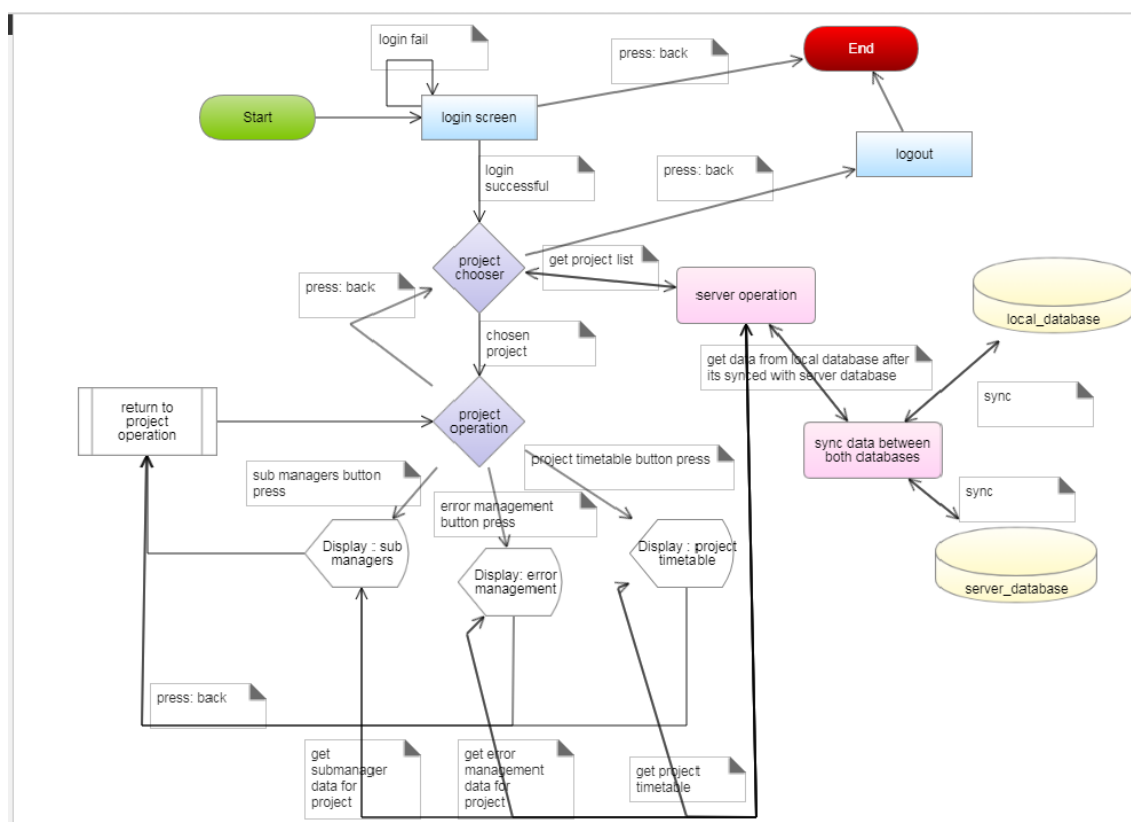


www.websequencediagrams.com

כמו שמתואר המנהל עבודה ניגש לאפליקציה, מתחבר ובוחר בפרויקט ובוחר לעדכן שדה כלשהו, ובשלב זה האפליקציה מאבדת חיבור אל השרת, האפליקציה דואגת לשמור את הפעולה בזיכרון המכשיר ומציגה למנהל עבודה הודעת הצלחה ותוספת שהשינוי יעודכן בשרת אחרי שחיבור אל האינטרנט יוקם מחדש, מצב זה מראה בצורה אידיאלית שקיפות והגינות כלפי מנהל העבודה ומקל על המנהל עבודה – כי במקרה אחר הוא היה צריך לשמור את הנתונים בצד בצורה

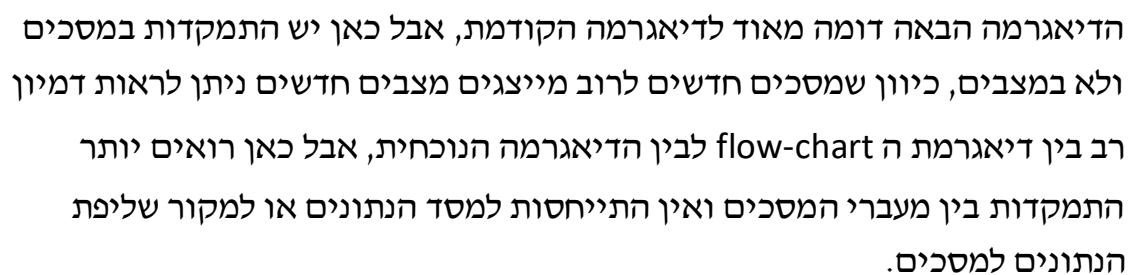
### Flow-chart

בתרשים זה אנו מציגים את מערכת המצבים משלב ההתחברות עד תצוגת המידע במסכים ואפילו עד היציאה מהאפליקציה והסגירה שלה



כמו שניתן לראות לאחר ההתחברות ישנו בורר מצב פרויקטים ולאחר בורר מצב הפרויקטים יש בורר מסכים שמפרט איזה מידע נרצה לראות על הפרויקט כל המידע הזה מתקבל מגוף כלשהו שמסנכרן את המידע בין הזיכרון cache המקומי לבין השרת, ולאחר מכן המידע נשלח אל המסכים הרלוונטים

### דיאגרמת מצבים למסכי הפרויקט



## 13



**מסך ההתחברות – אופציית ההזנה הידנית של משתמש וסיסמה במידה ויש**

**משתמש חדש להוסיף לזכרון המכשיר**



**במידה והמשתמש רוצה למחוק משתמשים מהזכרון מכשיר או לשנות את הסיסמה**

**המוזנת אוטומטית בתהליך ההתחברות, יש מסך שליטה לזה**



**לאחר מסך ההתחברות – בורר הפרויקטים, בחירת הפרויקט בו המשתמש ירצה**

**לעסוק**



### האופציות השונות שניתנות לאחר בחירת הפרויקט



### אופציית קבלני המשנה

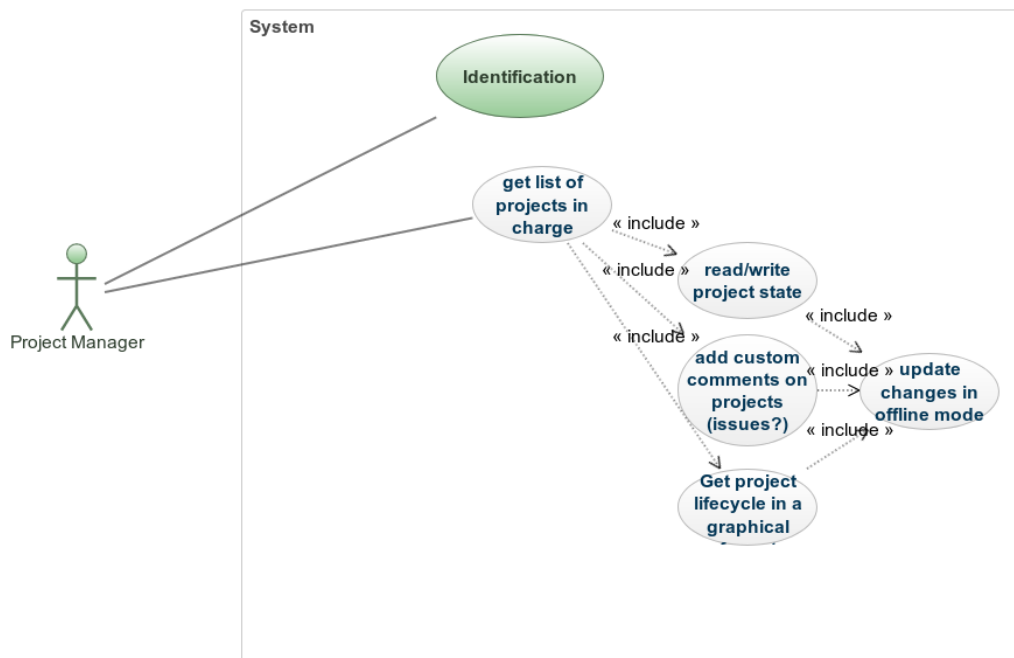


### אופציית הדיווחי תקלות



### Use-cases

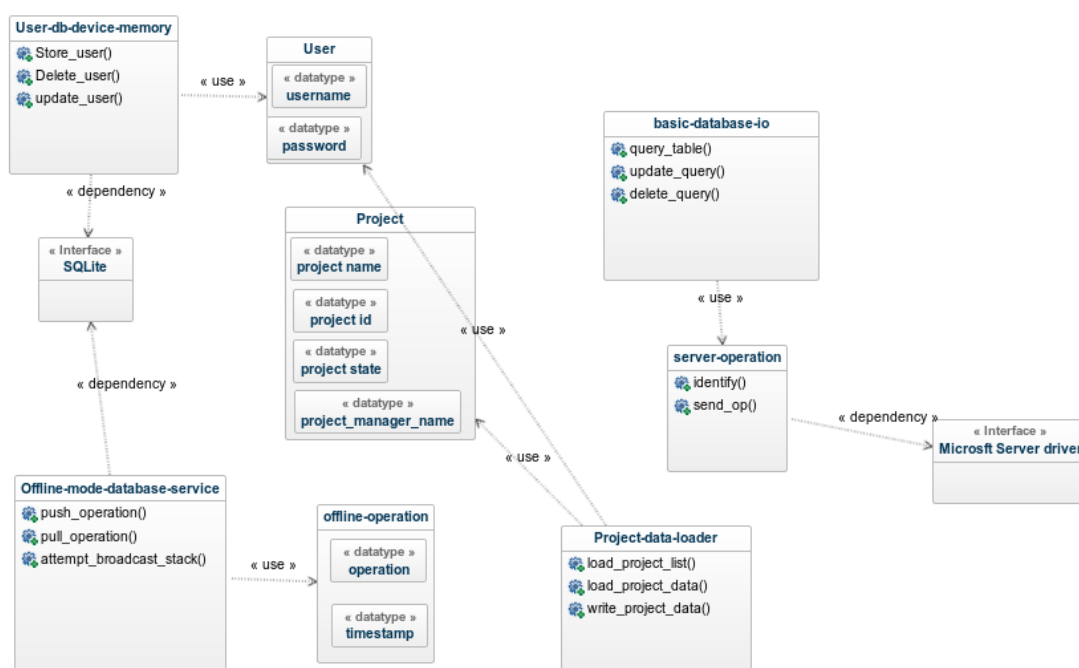
התרשים הבא מייצג את המנהל עבודה ואת השירותים שהוא יכול לקבל מהאפליקציה בשלב הנוכחי שלה, התרשים לא מייצג את המוצר הסופי כיוון שעוד לא ניגשנו אל מסד הנתונים ואנחנו לא יודעים איזה מידע נמצא במסד הנתונים אנחנו מראים רק חלק מהמידע שאנחנו מודעים שהוא יהיה נגיש אלינו והמנהל עבודה יהיה מעוניין לראות





## UML

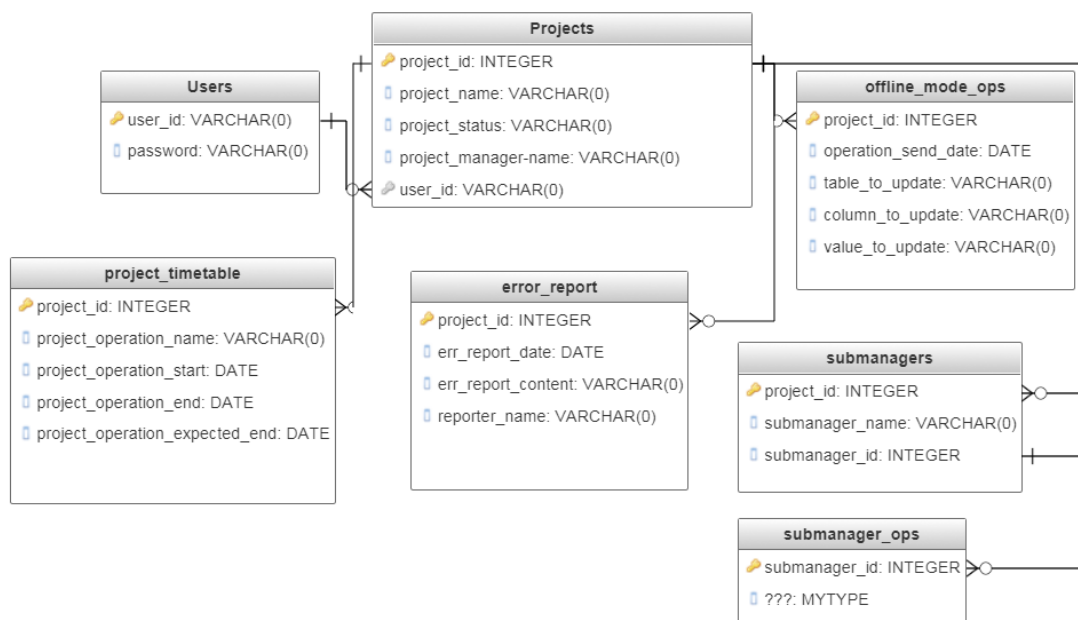
התרשים הבה מייצג את יחסי המחלקות שידועות לנו עכשיו בצד הלוגי, כמובן שקיימים מחלקות שמטרתם היא יותר גראפית (כמו ה Activity של Android או ה OpenSQLHelper של Java\Android או ה JDBC), אנחנו לא נתייחס אליהם אפילו שחלקם מתייחסים לחלק הלוגי של האפליקציה, אלא נחשוב כי קיים Interface בשם SQLite שמטפל בחלק המקומי של המסד הנתונים ו Interface בשם Microsoft SQL driver שיטפל בחלק המקוון של המסד הנתונים.



## טבלאות במסד נתונים

מסד נתונים לזכרון משתמשים שחוברו לאחרונה – בו נרשמים שמות משתמשים וסיסמאות של משתמשים אשר הצליחו להזדהות מול השרת, מקל על תהליך ההתחברות החוזר ונשנה מאותו המכשיר

מסד נתונים לזכרון cache של נתונים מצד השרת כדי לספק את התשתית לoffline mode (כלומר ישמרו נתונים כמו שמות פרויקטים ונתונים אחרונים שעודכנו, ופעולות על השרת שלא הצליחו להגיע אל השרת כתוצאה ממצב לא מקוון של המכשיר ישמרו במסד הנתונים הזה וימחקו רק לאחר הצלחת שליחת הנתונים אל השרת)



### ג. תכנון הפרויקט

פגישת הכרות עם הלקוח	24.8
פגישה נוספת על הלקוח כדי להבין את דרישות הלקוח	7.9
פגישות קבועות בימי רביעי עם צוות הפיתוח לצורך קידום הפרויקט, פיתוח וחלוקת עבודות (מעין scrum)	22.10 – הגשת הפרויקט
פגישה עם הלקוח בה נציג PoC של המוצר	סוף סמסטר א'

[ תאריך שיקבע על ידי הרכז ]	הגשת מוצר במצב ע"פ דרישת הרכז בסוף סמסטר א', זה כנראה יהיה מוצר אלפא
[ תאריך שיקבע ע"י הרכז ]	הגשת מוצר בטא למנהלי העבודה של הלקוח שלנו לצורך קבלת feedback לאפליקציה (כנראה בעיקר חווית משתמש)
שבועיים לאחר הגשת מוצר הבטא	נגיש מוצר בטא נוסף עם תיקונים ע"פ בקורות מהשחרור בטא האחרון
[תאריך שיקבע על ידי הרכז]	הגשת הפרויקט
[תאריך שיקבע לאחר סיום תהליך הפיתוח]	הגשת האפליקציה ללקוח והעלאה לחנות של גוגל

#### ד. טבלת סיכונים

#	הסיכון	חומרה	מענה אפשרי
1	עיקוב רב עקב תהליכי פיתוח שאנו חשבנו כי יהיו פשוטים אך קשים יותר בפעולה מול שרת הלקוח	4	קריאת הספרייה של התוכנה שמורצת ע"ג שרת הלקוח ושליחת מיילים במידה ולא נמצא פתרון בחיפוש ברשת
2	סיכון למידע השמור ע"ג השרת עקב באגים שיתרחשו בתהליכי הפיתוח	9	ניסיון למצוא סביבת פיתוח מקבילה לסביבת העבודה שבה המערכת שלנו תעבוד, או לבקש מהלקוח מעין "נישה" בטוחה בשרת
3	המוצר הסופי לא יספק את חווית המשתמש הרצויה של קהל היעד שלנו	2	המוצר ישוחרר בכמה גרסאות כמה פעמים במהלך חיי הפרויקט כדי לקבל feedback מצד קהל היעד שנוכל לשפר את חווית המשתמש בתהליכי הפיתוח ואף לקראת סיומה
4	גילינו לאחר פיתוח הבטא ולאחר הגשת האפליקציה כי חלק מהחבילות שנשלחות מכילות מידע רגיש בצורה גלויה	7	כיוון שאבטחת מידע היא התמקדות מאוד גבוהה בסביבה הארגונית, נודיע מיד ללקוח על הבעיה ונסה ככל שניתן ביכולתנו לפתור את הבעיה טרם פיתוח, אחרת פשוט נודיע למשתמש בצורה גראפית על הסיכון

#### ה. רשימת/טבלת דרישות

פורמט טבלת הדרישות יהיה לפי המקובל בארגון. להלן דוגמא:

#### טבלת דרישות (User Requirement Document)

מס' דרישה	תיאור
1	המוצר יכול להזדהות מול שרת הלקוח בצורה ייחודית כל משתמש לכל מנהל עבודה – תוכנה
2	המוצר יכול לתקשר עם שרת הלקוח כדי לשאוב נתונים ולכתוב נתונים על הפרויקט הנתון – תוכנה
3	המוצר ידע לעבוד במצב offline, כאשר לא קיים חיבור אל האינטרנט פעולות על השרת ישמרו בזכרון המכשיר וישלחו בהזדמנות – תוכנה
4	המערכת תהיה מודולרית על מנת שתוספים ופיצ'רים חדשים שהלקוח ירצה להוסיף בהמשך חיי הפרויקט לא יהוו מכשול לפרויקט – תוכנה
5	המוצר חייב לתמוך במערכת הפעלה אנדרואיד בגרסאות המקובלות לשימוש כיום (6 ומעלה) – חומרה/תוכנה
6	במהלך חיי הפרויקט רמת האבטחת מידע של המוצר חייבת לרצות את הלקוח כגוף ארגוני – אבטחה/תוכנה

### טבלת סיפורי משתמשים (User Case Stories)

1	בתור מנהל עבודה, אני יכול לשלוח נתונים לעדכן את השרת בנתונים חדשים מבלי להתייחס למגבלות המכשיר לחיבור לאינטרנט
2	בתור חבר בצוות ניהול החברה, אני יכול לראות את רשימת כל הפרויקטים בלחיצת כפתור פשוטה מבלי להכנס לדפדפן ולשירותים טכנולוגיים כמו vpn ו rdm כדי לראות את מצבי הפרויקטים על מסך הפלאפון שלי
3	בתור מנהל עבודה, אני יכול לראות תרשים המדמה כלי מאוד חזק בעולם ניהול הפרויקטים מבלי לחשב את הנתונים בצורה עצמאית – מכשיר האנדרואיד שלי מחשב אותם עבורי
4	בתור אחראי אבטחה המידע של הארגון, יותר פשוט מבחינתי להתייחס למידע שזולג מהאפליקציה מאשר להתייחס למידע שזולג מכל דפדפן או כל שירות בנפרד, כל ה"סיפור" נשאב למקור בודד
5	בתור מנהל עבודה, יש לי את היכולת לראות את הנתונים האחרונים שראיתי על הפרויקט בצד השרת ללא צורך בלשמור אותם בצד בצורה עצמאית, וללא תלות בחיבור שלי אל הרשת