

# ORION CONNECTION



# תוכן עניינים

	<b>מבוא</b>
<b>4</b>	
4	רקע
5	חקר שוק
5	Logitech ARX Control
6	Assistant Computer Control - ACC
7	Push2Run
7	Google Home App
8	לעומת הפתרונות הקיימים Orion Connection
9	חker פיתוח
9	הערכת אתגרים
10	מערכות הפעלה
11	שפות תכנות
11	פרוטוקלים
12	תקשורת IoT
13	תקשורת בין אנדרואיד לSERVER
13	Server Hosting
14	שליטה במחשב
14	Wake On Lan
15	UI
15	סקיצה
16	אפליקציה
22	תוכנת המחשב
<b>25</b>	<b>Detailed Design</b>
25	תקשורת בין המכשירים
26	תהליך החיבור - Pair
28	מבנה הפרויקט
28	השרות
29	תוכנת המחשב
30	האפליקציה
31	פרוטוקולים
31	פקודות מהSERVER

32	פקודות מחשב
32	פקודות מטלפון
33	מבנה הודעות
<b>34</b>	<b>רפלקציה</b>
<b>36</b>	<b>הוראות שימוש ומידע כללי</b>
36	Source Code: App, Server, Software.
36	Demo
37	תוכנת המחשב
37	דרישות מערכת
37	הרצה
37	שימוש
37	חיבור ראשוני
37	MAGIC
37	LOGGER
38	אפליקציה
38	דרישות מערכת
38	התקנה
38	שימוש
38	חיבור ראשוני
38	MAGIC
38	GROUP
38	ROUTINES

## מבוא

### רקע

כיום, יותר ויותר מוצרים בשימוש יומיומי בביטחון הופכים להיות "חכמים" מה זה חכמים? מוצר או חישון שניתן לשימוש וקלוט את נתוני אמצעות מכשיר טכנולוגי המתחבר בפרוטוקול תקשורת שונים למכשירי הבית החכם. כוים מוצרי בית חכם משתמשים על התעשייה וכוללים בהםן מערכות מיזוג, תאורה, רמקולים, טלוויזיות, מצלמות אבטחה, מנועלים ועוד.

עד היום, התחום העצום הזה כמעט ולא נגע במוצר אחד דומיננטי בחיה היום יומם שלנו - המחשב האישי. המחשב הוא אחד המכשירים הטכנולוגיים הכי מרכזים בחיננו ורובנו משתמשים בו כמה שעות טובות בכל יום בין אם לעבודה, לימודים או פנאי.

מי לא נתקל בסייעת המתחסלת של لكم מהספה הנוחה לעבר המחשב המחבר לטלוויזיה כדי לעזור או לשימוש בווילויים. או لكم מהሚיה כי שכחتم לכבות את המחשב. הרינו אנחנו בעידן הטכנולוגיה, אם אני לא צריך لكم מהמייה לכבות את האור, למה אי אפשר לעשות את אותו הדבר למחשב המשמש ברוב המקרים כמרכז בידור למיזיקה וטלוויזיה. בתור אחד שמשתמש במחשב לא מעט שעות ביום והמחשב אני יודע כי נתקלתי במספר לא זניח של מצבים בהם מחשב כמו צריך בבית חכם היה מאד עוזר.

האפליקציה של Orion Connection תפזר את בעיה זו. היא תביא את יכולת שליטה מוכנת ביכולה המחשב דרך טכנולוגיות הבית החכם. שליטה מוחלטת בווילויים, בכיבוי, הדלקה והציגת נתונים שונים על מחשב והכל מרוחק באמצעות הסマרטפונן בין אם דרך אפליקציה ייעודית או באמצעות העוזרת האישית.

## חקר שוק

כיום יש מספר פתרונות בודדים לשילוט מוחלט של המחשב מרוחק אף אחד מהם לא מספק נוחות שליטה וריבוי פיצ'רים.

### Logitech ARX Control

פתרון זה הוא אפליקציה שפותחה חברת Logitech. השימוש באפליקציה די נוח והתקינה פשוטה: התקינה תוכנה על המחשב (GHUB) או Logitech Gaming Software (LGSM) ואפליקציה הננתמכת גם ב-SO וגם ב-Android.

הפתרון מספק שליטה נוחה על ה-Media Keys. כמובן, הוא מאפשר שליטה על עוצמת סאונד המחשב, עצירה ומעבר בין שירים. בנוסף הוא מספק מידע מלא על טמפרטורת רכבי המחשב ואחדו השימוש בהם בזמן הנוכחי. מן האפליקציה בטלפון ניתן לפתח משחקים על המחשב, לשנות DPI והגדירות נוספות של העכבר.

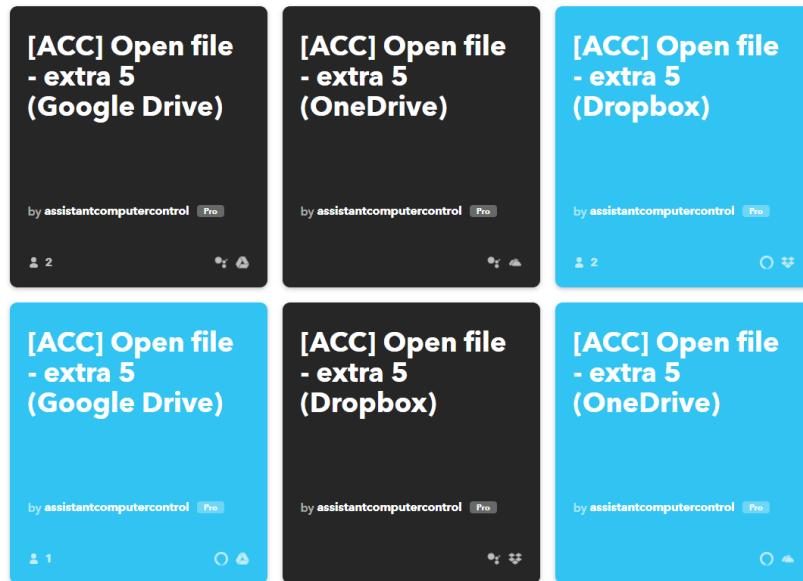
החסרונות של פתרון זה מאוד בולטים - התנאי לשימוש באפליקציה הוא מציאות רכיב Logitech תומך כגון מקלדת או עכבר, ללא רכיב מתאים האפליקציה לא תפעל. בנוסף, השליטה על המחשב תליה בחיבור רשת מקומי, ככלור הלקוח (טלפון) והсер버 (המחשב) חייבים להיות מחוברים לאותה הרשת, אחרת לא יהיה ניתן להשתמש באפי פיצ'ר. פתרון זה די יישן ולא מקבל יותר תמיכה מהפותחים מאז הרבעון הראשון של שנת 2016, ככלור אין עדכוני גרסה של פיצ'רים חדשים ותיקוני באגים קיימים המפריעים לשימוש השוטף.



## Assistant Computer Control - ACC

פתרון הממוש בקוד פתוח הנוטן שליטה על פונקציות שונות בעזרת העוזרת הקולית של גוגל או אמזון. על מנת להשתמש בתוכנה יש להתקין תוכנה אחת בלבד על המחשב ולהתחבר ממנה לחשבון IFTTT וחשבון ענן נוסף לבחירה - DropBox, OneDrive, Google Drive - .  
בעזרת קישור פקודות קוליות מה-IFTTT התוכנה כתובת למסמר טקסט המאוחסן באותו שירות ענן ובעת קריאה של אותו מסמר המחשב מבצע את הפעולה הנבחרת.  
פתרון זה מביא את עקרונות הבית החכם למחשב האישי עם יתרון משמעותי של ממש ישר עם העוזרת האישית בטלפון ונוטן גישה למספר די גדול של פקודות שניית לביצוע. אך, פתרון זה לוקה בחסרונות רבים, אין ממש פידי במכשיר הלקוח כר שכל העברת פקודה תהיה ארוך ורך ממש קול. בנוסף, התוכנה של תוכנה זאת מצריכה שלבים רבים של חיבור חשבונות והוספה פקודות ב-IFTTT באופן ידני על כל פקודה.

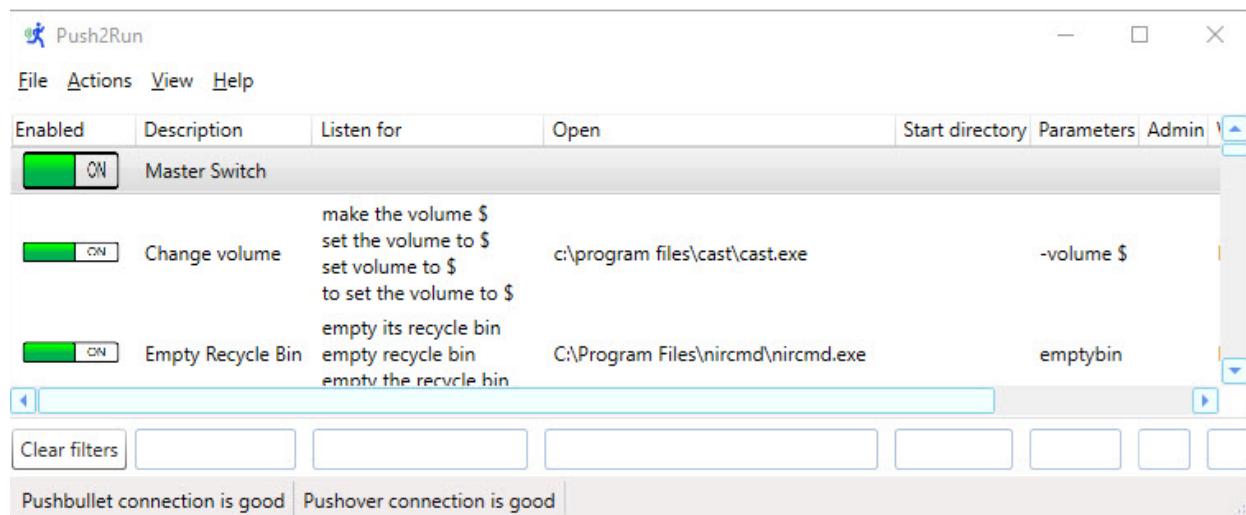
### assistantcomputercontrol's Applets



## Push2Run

פתרון זה דומה מאוד ל-ACC, הוא מKENה שליטה מלאה דרך העוזרת האישית בעזרת קישור IFTTT ושירות הנקרא Push Bullet המאפשר שליחת הודעות בין השרת ללקוח. פתרון זה מסובב מבחינות תחילה העבודה איתנו. הוא מספק רק את התשתית לאחר הפתרון וכל השאר צריך ליצור לבד. כמובן, את הפקודות ב-IFTTT ואת הפקודות בתוכנה עצמה (פקודות של CMD).

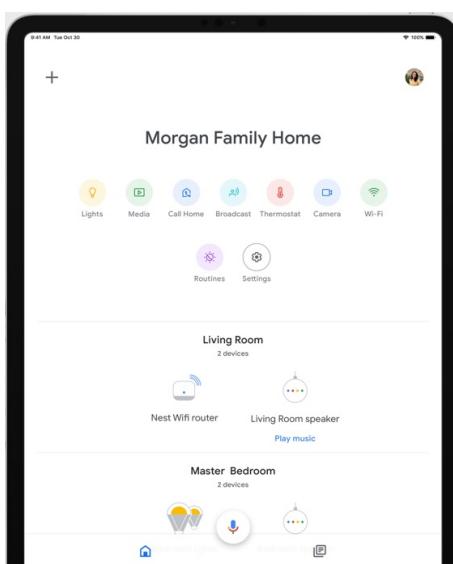
היתרון המאוד ייחודי שהפתרון מציע הוא יכולת השליטה הבלתי מוגבלת, מכיוון שאין תוכנה פתוחה, כמובן ניתן לכתוב לבד ולהוסיף כל פקודה אפשרית, רק צריך לחקור ולדעת את ה-Syntax לכך.



## Google Home App

אפליקציה זו היא אפליקציית iOS מתוצרת גוגל אשר מאפשרת שליטה במגוון מוצרי בית חכם דרך מקום אחד. תוך כדי היא אפשרות פיצרים של שליטה על ידי העוזרת הקולית של גוגל, יצירתلوحות זמינים וKİצורי דרך קוליים. אמנם מטרת האפליקציה במקרה זה שונה מ-Orion Connection אך, ישנו

מספר קווים דמיון ממשמעותיים. עיצוב ה-UI הנוח להפליא, שליטה על מגוון מכים ויצירת Routines המאפשרים שליטה בעזרת אוטומציות בעיצוב אישי.



## לעומת הפתרונות הקיימים Orion Connection

פתרונות הקיימים בשוק מציעים בחירה בין עם שליטה מוחלטת בפונקציות שונות בעזרת העזרת הקולית בלבד או שליטה במבחן מצומצם של פונקציות שמוגבל לשילטה ברשות המקומית בלבד ובלבד. הפתרון שלי יציעו לנו אפשרות נוח לשילטה מוחלטת בכמות די גדולה של פונקציות ללא הגבלה לרשות מקומית. ובנוסף תבחן אופצייה לשילוב של עזרת הקולית של גугл לתוכה פונקציות השילטה. הAKERON המרכז של הפתרון שלי הוא להביא את יכולות השילטה לכל אדם גם אם הוא אינו מבין כמה גדול במחשבים ובטקשורת, בדיק כמ שAPPLICATIONS שליטה בבית חכם עושות.

## חקר פיתוח

### הערכת אתגרים

הפרויקט שלי עוסק בעיקר בתחום התקשרות בין הטלפון לבין המחשב בין אם ברשת פרטית או ציבורית. לכן, אני מעריך שהאתגר העיקרי בו אני אתקל יהיה עניין התקשרות בין המכשירים. ליצור פרוטוקול בו שליחים פקודה למחשב מרוחק והוא צריך לפעול בהתאם, גם אם הוא אפיו לא דלוק. הפרוטוקול יצרר לוודא כי הפקודות אכן התקבלו וובוצעו כמו שצריך גם אם הפקודה היא כיבוי הרםטי של המכשיר.

בנוסף, הפתרון שלי מורכב מAPPLICATION שרת וAPPLICATION ללקוח כך שהלקוח נמצא על מערכת הפעלה Android. אין לי שום רקע בתכנות ב-Android لكن סביר כי אתקל בקשיטים בזמן הפיתוח הקשורים לעקרונות התכנות במערכת הפעלה זו.

## מערכות הפעלה

הפרויקט שלי הוא פרויקט בשלוש חלקים - סרבר, תוכנה על המחשב ותוכנה על הטלפון لكن הוא יאלץ לפעול בשלוש מערכות הפעלה.

מערכות הפעלה הפופולריות ביותר למכשירי הטלפון הם: poidroid של Android ו- iOS של Apple. כרגע שנכנן לסוף הרביעון האחרון של שנת 2020 אנדרואיד מפעילה כ- 72.92 אחוזים ממכשירי הטלפונים ברחבי העולם ו- iOS מפעילה כ- 26.5 אחוזים. ה- 0.58 אחוזים הנותרים מפעילים מערכות הפעלה כמו - Windows Phone ו- KaiOS ועוד כר' שם זמינים.

על מנת לתכנת אפליקציה תואמת למכשירי אפל דורשת למעשה ייעודית לשפת תכנות לפיתוח אפליקציות ב- iOS ושמה Swift. אין לי שום ידע קודם בשפת תכנות זו ולכן מפהת שיקולי קושי ותרתי על ניסיון פיתוח למכשירי אפל. זאת בנוסף לשיבת כי במהלך השנים האחרונות אף הקשישה את כניסה של אפליקציות צד שלישי (אפליקציות שלא נמצאות בחלון האפליקציות שלהם) لكن בחרתי להימנע מסיבור מיותר.

מערכת הפעלה Android מגוונת בגרסאותיה. כיום יותר מחצי המכשירים עליהם מופעלת מערכת ההפעלה Android היא ב-3 גרסאות האחרונות: 9, 10 ו- 11. מכיוון ש- 11 עוד חדשה בשוק אני נאלץ לזרור על פיתוח ייעודי אליה מפהת חומר יכולת בדיקה וריבוי באגים ועקבונים שבועיים מצד Google או המפתחות האחרות.

שוק מערכות הפעלה למחשבים דומה לשוק מערכות הפעלה בתחום מכשירי הסלולר. Windows של Microsoft פועלת כמערכת הפעלה ליותר מ- 76 אחוזים, X OS של Apple הפעלת ביותר מ- 17.5 אחוזים. באחדים הנותרים יש את - Chrome OS, Linux ועוד.

אנו שיקולי פיתוח מצד הלקוון פעולים גם כאן. מערכת הפעלה של Apple דורשת למעשה של שפה נוספת שמיון שהיא מהוות פחות מחמישית בmachines ברחבי העולם הלמידה הנוסף אינה כה משתלמת. בנוסף לשיקולים אלו ולשיקול של קושי רב בפיתוח אפליקציות על מחשבי Apple, אין לי מכך הרוץ את מערכת הפעלה זו ولكن היא נשלה.

מערכות הפעלה לשרתים הן מגוונות בתוכנותיהם, ישנים כל מיני שירותים המספקים שירותי מובסים על מערכות הפעלה שונות, מכיוון ש- Linus היא המערכת הפעלה המובילה בתחום אין פה ממש התלבטות. הסיבה לכך היא כי Linus היא מערכת הפעלה עם המון תמייה ודוגמאות שניתן לקחת השראה ולימוד מהם, זאת לעומת שאר המערכות שהן יותר מקצועיות וספקות שירותי סרברים מקצועיים.

על פי המחקר הבנתי כי הפרויקט שלי יירוץ על - poidroid מצד האפליקציה, Windows מצד המחשב ו- Linus מצד השרת.

## שפות תכנות

אני בחרתי לכתוב את הפרויקט בשלוש שפות: XML, Java, Python. Python היא שפה נוחה וובאה לכתיבה של תקשורת בין סוקטים, בנוסף יש לי אותה ניסיון רב בכתיבה של פרויקט תקשורתי אחר, لكن בחרתי בה להוביל את עניין הלוגיקה והתקשורת.

בשל כתיבת אפליקציה מתאימה لأنדרואיד נאלצתי להוסיף עוד שפות, שפות מתאימות לתוכנות אנדרואיד. השפות המומלצות לתוכנות בסביבת אנדרואיד הם C# ו-Java כאשר Java מוגדרת כהשפה הרישمية בתוכנות אנדרואיד. בשל ניסיון רב שיש לי ב-Java ובשל התמיכה הרבה שיש אליה בנושא האנדרואיד, בחרתי בה.

שפת XML היא שפה נדרשת בתוכנות בסביבת פיתוח של אנדרואיד, בעזרתה ניתן לפתח GUI ולהגדיר הגדרות בסיסיות לאפליקציה.

בנוסף, אני בחרתי בשפת YAML, שהיא שפה יעילה המתמחקת יישירות עם Python. היא שפה די נוחה שנitin להשתמש בה בפייטון באמצעות קישור פשוט, יש לי ניסיון עמה, لكن בחרתי לעצב אליה את ה-GUI של תוכנת המחשב שלא מצריכה יותר מדי סיבור גרפי.

## פרוטוקלים

הפרוטוקולים הנפוצים ביותר בעולם הם - UDP ו-TCP. כאשר לכל אחד יש חסרון ו יתרונות משלו, כאנו נשאלת השאלה במה לבחור או האם כדאי לעשות שימוש.

UDP User Datagram Protocol הוא פרוטוקול תקשורת שהשימוש העיקרי בו הוא יצירת חיבור מהיר בין אפליקציה לאינטרנט עם מידע כי יכול להיות שמייד ואבד בדרך ולעולם לא הגיעו ליעד. השימושים מאד ברורים של הפרוטוקול הם: VoIP ו-Video Streaming - Video Streaming.

דילוי בין הלקחות, ופחות משנה אם באמצעות שנייה התפספה בזידאו כל עוד זה מהיר.

TCP Transmission Control Protocol הוא פרוטוקול המגדיר את הסטנדרטים בנוגע ליצירה ותחזוק שיחה אינטראקטיבית בין תוכנות המחליפות פקודות של מידע. TCP הוא פרוטוקול אמין, המתחזק החלפת מידע ללא איבודו לעומת UDP. TCP יותר איטי בשל פיצ'ר המאפשר ביחסון בעת העברת מידע וקבالت הפקטה במלואה. כאשר רוצים לבחור בין הפרוטוקולים צריך לקחת בחשבון את ההתלבבות בין המהירות שמספק ה-UDP לעומת שירות של העברת מידע אמין שהוא TCP מספק.

מעבר לכך לכל פרוטוקול יש פיצ'רים ייחודיים לו בשל המבנה הייחודי של כל אחד מהם. לדוגמה: אף רוק ב프וטוקול UDP ניתן לשולח הודעה Broadcast לכל מכשירי הרשת הפרטית - NAT בערוך מסוים ולפרוטוטוקול מסיים.

## תקשות IoT

תקשות בין רכיבי בית חכם זה נושא עצום המכיל בתוכו ענפים של סוגי פרוטוקולים ורכיבים שונים הפעילים בתחום הרשת. העיקרון המפתח המוביל את תחום הבית החכם הוא המעבר בין המכשיר אל המשתמש, בין אם ישירות או דרך תווות עזר.

יש כמה שיטות, להלן השיטות הנפוצות ביותר למכשיר IoT:

- **חיבור - מכשיר למכשיר:**

חיבור עצמאי בין מכשיר למכשיר באמצעות פרוטוקולים כמו: Bluetooth, Z-Wave, ZigBee. דרך חיבור זאת נפוצה בעיקר בתחום המכשירים הליביים שלא מצריכים הזרמת מידע לכמה מכשירים בונוסף, עיקרונו האבטחה של דרך חיבור מכשיר למכשיר מאוד פשוט מפני שבין המכשיר לבין המשתמש יש תקשורת בمعالג סגור אך ורק ביניהם ללא צורך במעבר במכשירים אחרים.

- **חיבור - מכשיר לענן / לרשת:**

חיבור כל מכשיר יעשה על ידי חיבור לשירות אינטראקטיבי חיצוני המספק החלפת נתונים ושליטה בהעברת ההודעות בין הצדדים. לרוב הפרוטוקולים המומומשים בשיטה זו הם: cellular, Ethernet, Wif, Ether.

- **חיבור - מכשיר ל-gateway:**

חיבור זה מוסיף שכבת מעבר לחיבור מכשיר לענן / לרשת. המכשיר עצמו מחובר ל-gateway ובכך יוצר מעבר נתונים: מכשיר -> gateway -> לרשת. משתמשים בשיטה זו במקרים בהם יש התערבות מצד התוכנה במימוש אפליקציה כמו סמארטפונים או מכשירי IoT שפועלם כמעבר ביןיים בין המכשיר לרשת.

## תקשרות בין אנדרואיד לשרת

התקשרות בין המכשירים קורת באמצעות פרוטוקולים שונים, רוב ה프וטוקולים האלה בסיסים מ모ומשים על ידי UDP או TCP. רוב אפליקציות האנדרואיד מתקשרות דרך שרת HTTP ודרך מעבירות את המידע הנדרש בצורה כזו שברוב המקרים השרת משתמש אך ורק כגשר בין הצדדים וublisher פקודות בין המכשירים (בין אם זה טלפון לטלפון או מחשב לטלפון).  
ניתן להשתמש בפרוטוקול UDP ו-TCP כמו שהם ללא שכבות על מה שלפעמים מקל על התקשרות והשימוש עצמו.

## Server Hosting

בין אם בוחרים ב프וטוקול TCP פשוט או שטח מוממשים תקשורת דרך HTTP, על מנת להגיע למצב שבו האפליקציה הטלפונית תಡע לתקשר עם השרת מכל מקום צריך להריץ סרבר בראש ציבורית פתוחה עם IP ידוע. ביום יש המונן שירותים בהם ניתן לשכור בכספי או בחינם חלקת עיבוד במחשב סרבר ובכך להריץ את הסקריפט המותאם לך. לשרתים כאלה קוראים Backend Servers, בהם השרת הוא כמחשב לכל דבר המריץ קוד בשפת backend.

תמיד יש את האפשרות אומרת כי להריץ את השרת על המחשב האישי בבית. אך, עם הנזונות שהפתרון מביא, הוא מביא עימיו כמה חסרונות לא זניחים: לא ניתן להריץ את השרת 24/7, יש האמורים כי הרצת סרבר בראש הפרטיה של הבית יכול לגרום לביעיות אבטחה חמורות. לכן, הפתרונות המוצעים בשוק הם: Google Cloud, Microsoft Azure, AWS, LINODE.

## שליטה במחשב

השליטה במחשב דרך תוכנה - סקריפט כתוב מראש, היא דבר לא מסובך. מערכת הפעלה Windows מספקת מגוון אופציות על מנת להציג את השליטה הרצויה. בין אם באמצעות פקודה ב-CMD הוא בין אם עובדים ישרות עם קבצי DLL מתאימים.

על מנת לשנות בהגדרות "ההערות" של המחשב (כיבוי, שינוי, הפעלה מחדש וכו....) השימוש בפקודות פשוטות מה-CMD הוא הפתרון הכי נוח, על הפקודות הפחותות ניתן להוסיף מגוון רחב של פרמטרים שיאפשרו טוימר של ספירה לאחר מכן לפקודה או הודעה...

על מנת להפעיל פקודות CMD דרך תוכנת-ho Python ניתן להשתמש בספרייה os המובנת בתוך פיטון:  
`os.system("shutdown /s /t 30")` פקודה זו תכבה את המחשב (הפרמטר s) ותפעיל טוימר של 30 שניות (הפרמטר t עם מספר 30) עד לביצוע הפעולה.

למרות המגוון הרחב של הפקודות הנitinנות לביצוע מה-CMD, חסירה פונקציה אחת הנקנכת בקטgorיה זו - נעלית המחשב. האילוץ להשתמש ב-DLL מיוחד בשbill פעה זה והופיע, מתוך ספריית ctypes של Python ניתן להפעיל כל קובץ DLL שקיים במערכת או לחלוין DLL מותאם אישית. במקרה זהה, ההרחבה מציעה פונקציה מובנת לנעלית המחשב באמצעות DLL: `ctypes.windll.user32.LockWorkStation()`

על מנת לשנות בפקודות הקשורות באופן ברור עם השימוש של המשתמש במערכת הפעלה, צריך לדמות את המשתמש בקוד, כלומר, ללחוץ על מקשיים ספציפיים במקלדת על מנת להشتיק, להגביר, להנמיך, לעצור מדיה ועוד...

על מנת לבצע דמיון של לחיצה על מקש דרך הסקריפט פיטון מציעה מגוון רחב של ספירות תומכות בפייצ'ר. חלקן נוחות יותר וחלקן פחות. ספריית `tkinter` לעומת זאת מושפעת מפתרונות מספקת את השימוש הינו, עם קבועים תוכנתיים מובנים בספריה על מנת להפעיל כל מקש מיוחד. דוגמת מקשי המדיה:

`sim_keyboard_press_(Key.media_volume_up)`

## Wake On Lan

הודעה אינטראקטית מיוחדת הנקראת גם Magic Pocket המועברת באמצעות חיבור קוו (Lan), ההודעה מאפשרת הדלקה מלאה של המחשב על ידי האינטראקט בלבד. בדרך כלל משתמשים בטכנולוגיה זו בראשת המקומית על ידי שימוש במכשיר אחר באותה רשת מקומית כמו טלפון או מחשב אחר השולח את ההודעה לראש. אך, קיימת גם אופציה למימוש תוכנה גם דרך רשת חיצונית זאת באמצעות שידור ו송 של ההודעה או באמצעות שירות WoL gateway.

## UI

תחום ה-User Interface מאוד חשוב לי, מבחינה אסתטית ונוחות, בתור מעצב גרפי אני שואף להגיע לעיצוב נוח, מודרני ויפה מבלי להסתבר יותר מיד. ו'יכול' "לעשות" את האפליקציה או "לשבור" אותה لكن המחבר בקשר נושא ישב בסדר העדיפויות מבחינת מבנה הפרויקט כתלות במרקם ה-UI.

ההתלבטות הראשית לגבי עניין ה-UI היה בתוכנת ה-Server. מכיוון שבפיתוח בסביבת אנדרואיד יש שפת עיצוב ושמה XML המשמשת ליצירת UI דינוח. XML די פתוחה ומאפשרת הגדרות מותאמות אישית, בנוסף לכך היא בנויה ישירות לסליבית הפיתוח של גугл - Android Studio מה שמאפשר יצירה UI באופן פיזי ללא כתיבת קוד, למורת יתרון משמעותי המשפיע מכיוון שבאזור כתיבת הקוד ניתן ליצור UI יותר מרשימים ואסטטי.

לעומת זאת, בסביבת הפיתוח ב-thonpython לא קיימות הרבה אפשרויות מעבר להרחבות כגון pygame...etc. לכן ההתלבטות פה הייתה יותר קשה, ולאחר מחקר נשארתי עם שני אפשרויות - Unity ו-C#.

Unity זו שפת עיצוב שיש לה אינטגרציה מלאה עם פיטון, יש לו ניסיוןアイテ והיא דינוח, החישרון המאוד גדול שלא הוא חוסר בתמייה ומקורות מידע ברשות.

לעומת זאת פיתוח UI בעזרת C# זו אפשרות מאוד נוחה כיון שהתמייה בהרחבות הפיתוח הם אינטגרטיבים, אך עם זאת מגע החשرون, סיבור, על מנת להשתמש ב-C# בפרויקט פיטון יש ליצור חיבור לא פשוט בין מסגרת ה-UI ומסגרת הרשות והלוגיקה.

ההחלטה הסופית הייתה ל选取 עם Unity, מפה את רצון לשמר על פשטות בשל UI מצומצם בכל מקרה בתוכנת המחשב.

## סקיצה

המטרה העיקרית של הפרויקט היא ליצור נוחות אצל המשתמש כמו אצל אפליקציות בית חכם אחרות. מתחת למשתמש מוצר שהוא יכול להבין ולהשתמש ללא כל צורך במידע טכני וטכנולוגי.

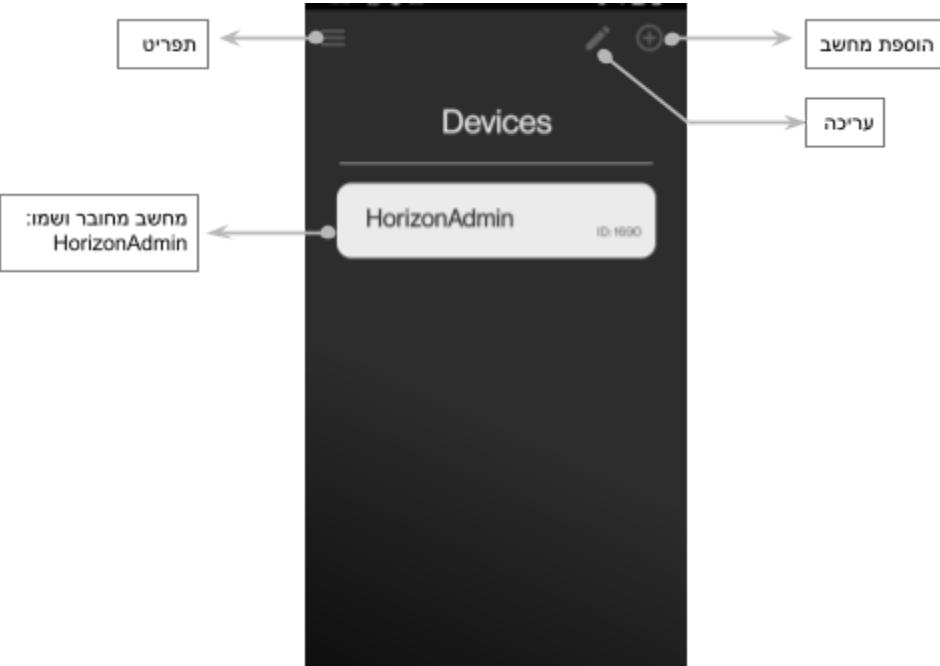
את הסקיצה עיצבתי בתוכנת XD:

**אפליקציה****מסך הראשי:**

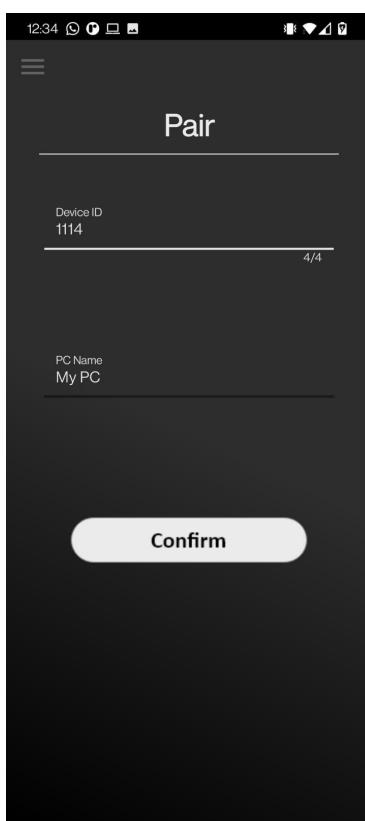
יכול את רשימת המחשבים המתחברים.

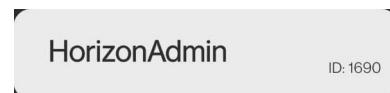
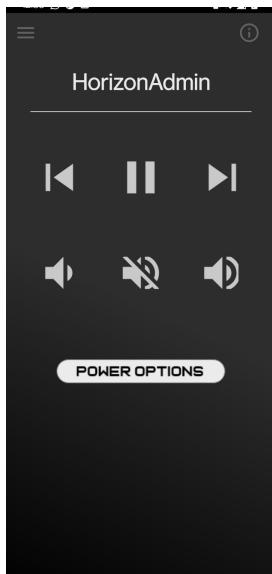
המסך מאפשר:

- להוסיף חיבור חדש
- לעורוך את החיבורים הקיימים
- לגשת לחיבור קיים

**מסך ה"כיוול":**

מטרתו להעביר את תהליך הכיוול בין מחשב לטלפון ומגעים אליו דרך לחיצה על כפתור הפלוס במסך הבית. המסך מכיל בתוכו עמודה למילוי PIN ייחודי היוצג על המחשב הרצוי. בנוסף, אפשרות לקרוא לו בשם.





### מסך השליטה:

מסך השליטה הוא אינדיבידואלי לכל מכשיר ומגיעים אליו דרך לחיצה על אייקון

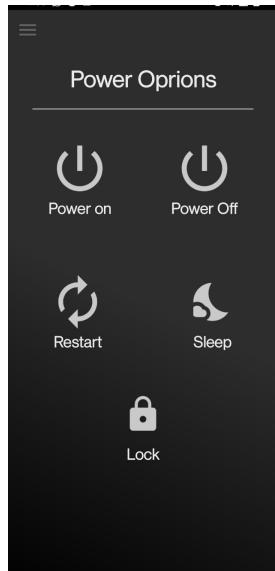
המחשב - למשל:

לאחר הלחיצה מגיעים למסך השליטה. הוא מכיל בתוכו שלושה אזורים:

- אזור השליטה בהגדרות ה"כח" - מכיל כפתור היבול למסך שליטה נוסף.

אזור השליטה בהגדרות המדיה - שליטה מלאה בווילום, העברת תכני מדיה ועיצובם.

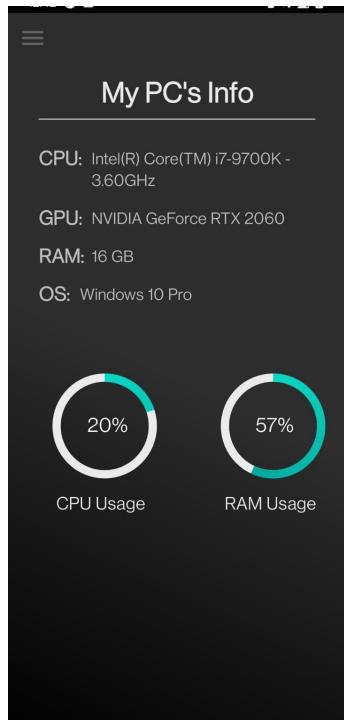
- אזור המידע - מכיל כפתור היבול למסך מידע נוסף לגבי המחשב.



### מסך ה-"Power Options":

מסך זה מכיל בתוכו אפשרות שליטה נוספת נוספת במצב הערות של המחשב.

הוא מציע בחירה בין - שינוי, נעלמה, אתחול, כיבוי והדלקה.

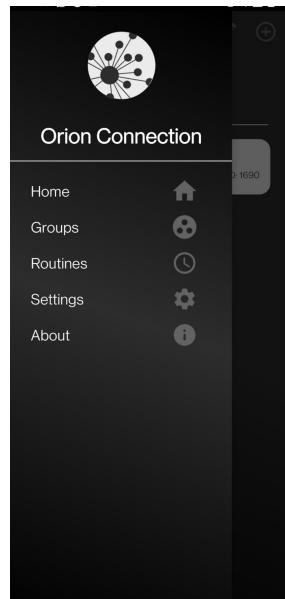
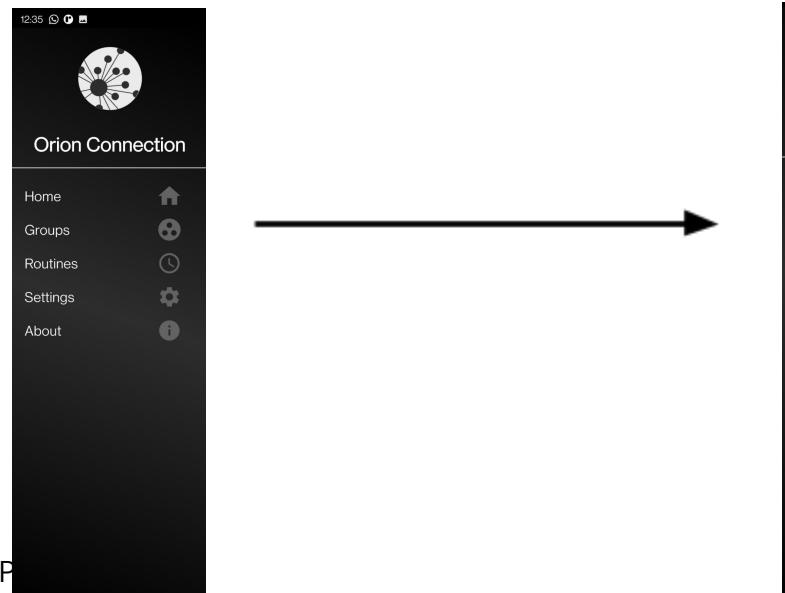


**מסך מידע נוסף על המחשב:**  
מסך זה מכיל מידע כללי על המחשב המחבר. טמפרטורות של רכיבים ומידע כללי על הרכיבים המרכיבים את המחשב.

- מידע על המעבד.
- מידע על הכרטיס מסך.
- מידע על ה-RAM.
- מידע על מערכת הפעלה.
- מידע חי על:
  - שימוש במעבד.
  - שימוש ב-RAM.

#### התפריט:

לכל מסך יש אפשרות לפתוח תפריט צד באמצעות כפתור: ☰ הממוקם בפינה השמאלית העליונה של כל מסך. התפריט יכול לאפשר חזרה למסך הבית (על ידי לחיצה על הלוגו) וכפתורי Routines | About |.



**פרק המידע:**

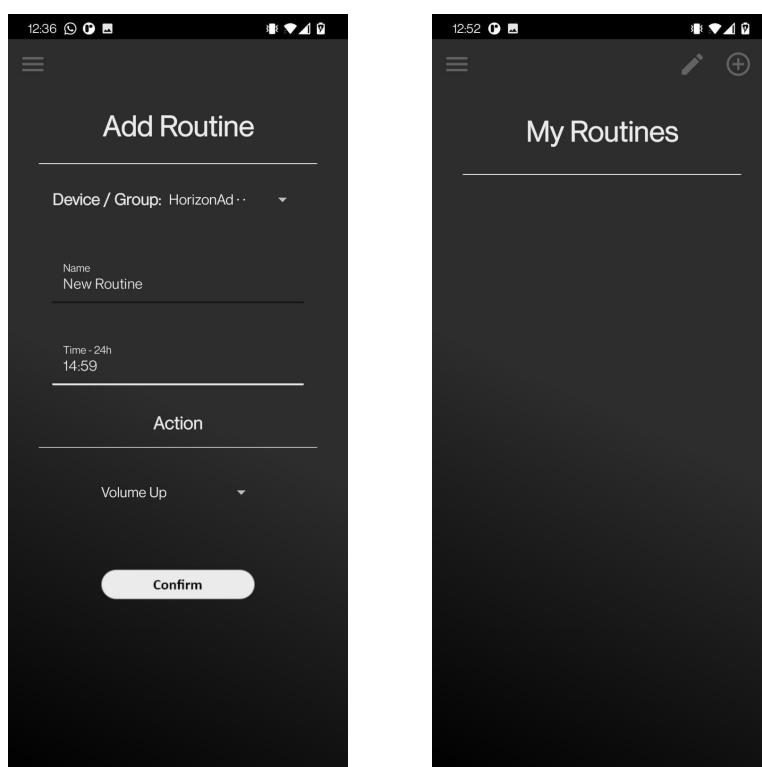
יכיל בתוכו מידע כללי על המפתח ועל הפרויקט.

**פרק ה"routines":**

בפרק זה, ניתן להגדיר תזמוןים שונים לפקודות או לקבצי פקודות. לדוגמה, ניתן להגדיר ששבועה 00:14:00 המחשב יעצור את המוזיקה וישתיק את הווילום. לכל routine ניתן להגדיר שם מסוון.

בפרק זה ניתן:

- להוסיף תזמון חדש
- לעורר את התזמוןים הנוכחיים
- לגשת לתזמוןים הנוכחיים



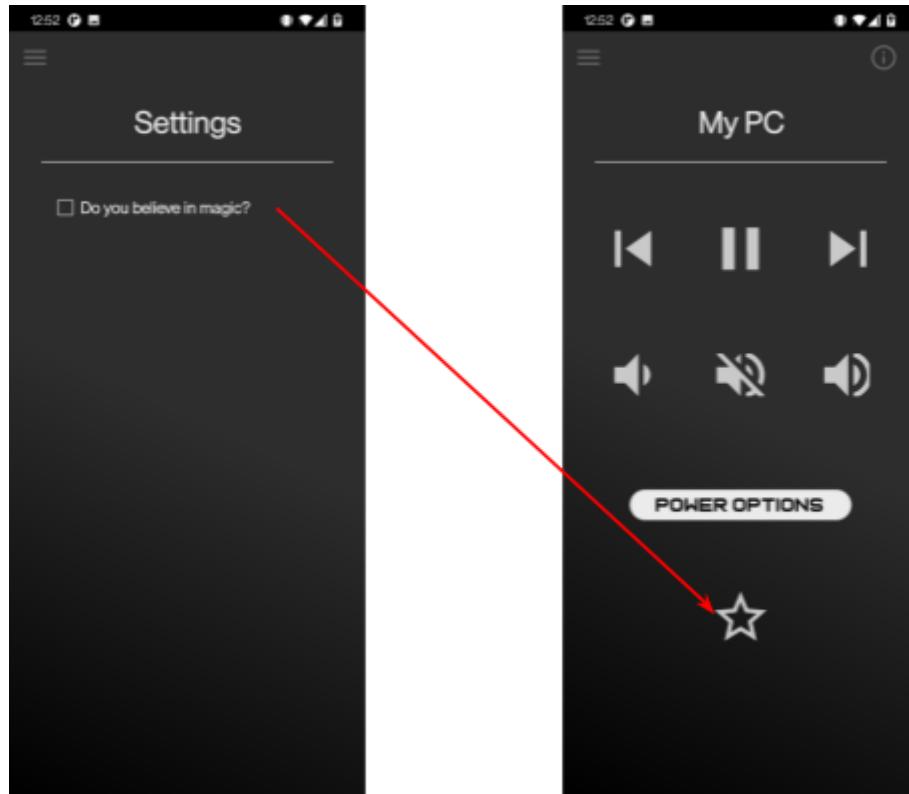
פסק הגדירות:

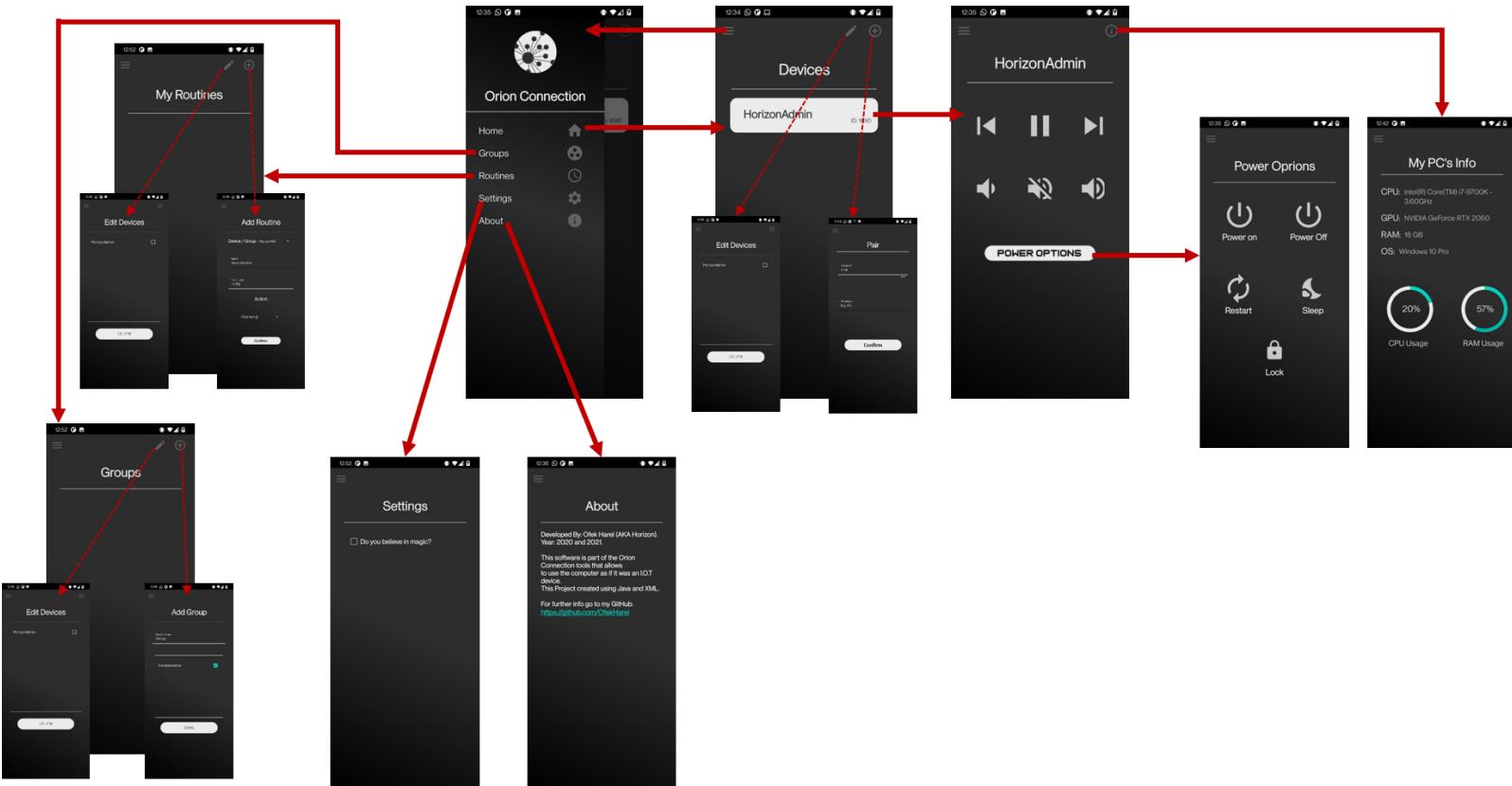
מכל הגדרה אחת:

- Do you believe in magic?

הגדירה זו תאפשר כפתור שטחתו כרצון המשתמש.

הגדירה זו קיימת כדי לוודא כי רק אחרי שהמשתמש הגידור בתוכנת המחשב כרצונו, הוא יפעיל את ההגדירה.



תרשים זרימה של מסכי האפליקציה:

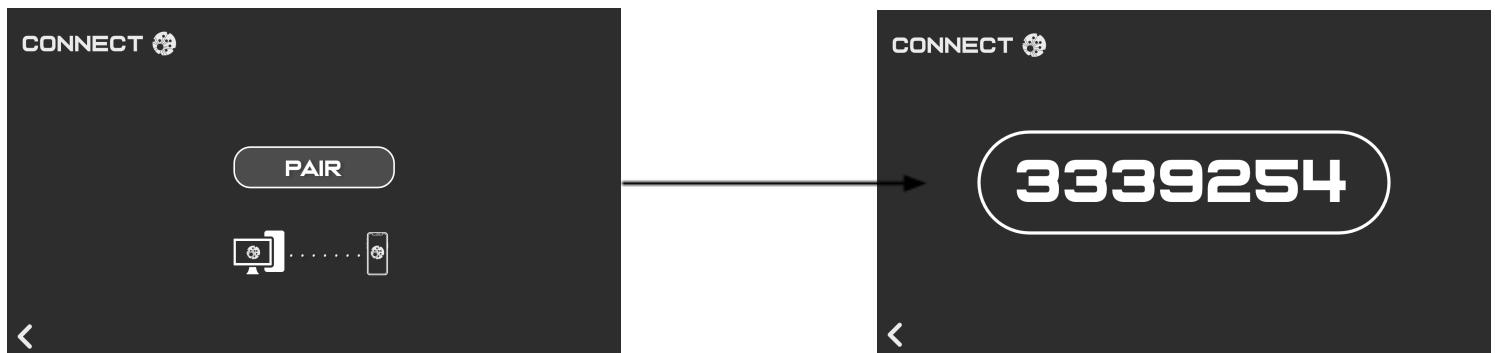
**תוכנת המחשב****המסמך הראשי:**

המסמך הראשי יציג ניווט בין כל הפיצרים של התוכנה - חלון התחברות, חלון לוגים וחלון מידע כללי.

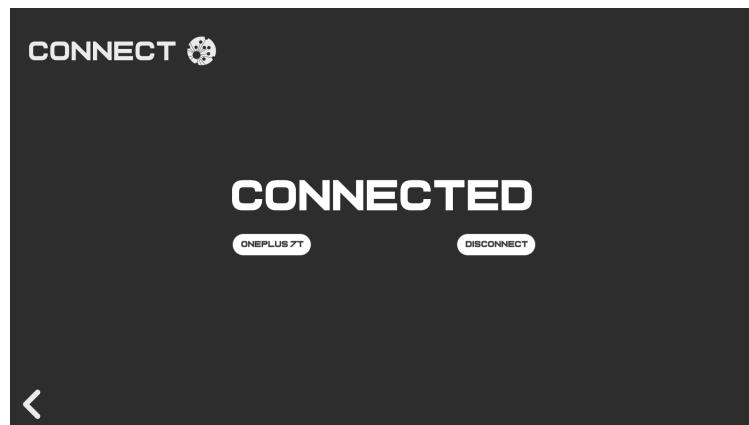
על גבי כל מסך יש כפתור חזרה, שייחזיר את המשתמש אל המסמך הראשי.

**מסך התחברות:**

בעת התחברות מכשיר חדש יש להיכנס למסך התחברות, אם המחשב אינו מחובר לטלפון אז תהליך הcoil הראשוני יחל. בעת לחיצה על כפתור ה- *pair* על המסך מופיע מספר PIN מיוחד המשמש להתחברות בין הטלפון למחשב.

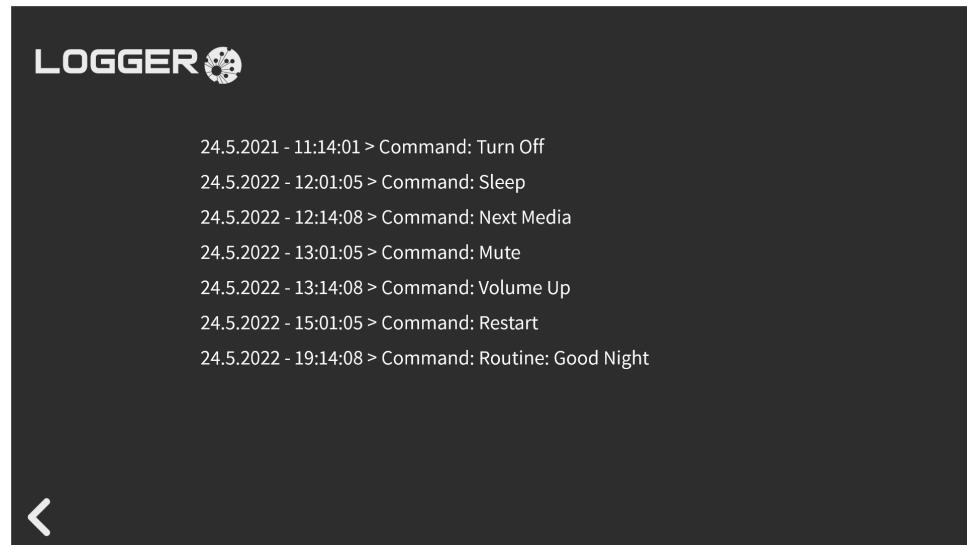


אם המחשב כבר מחובר, אז מסך עם מידע על גבי הטלפון המחבר + אפשרות ניתוק.



מפרק הלוגר (logger):

מפרק הלוגר יוכל בתוכו רשימה מלאה של כל הפעולות שהטלפון המרוחק ביצע על המחשב עם חותמת זמן ותאריך.



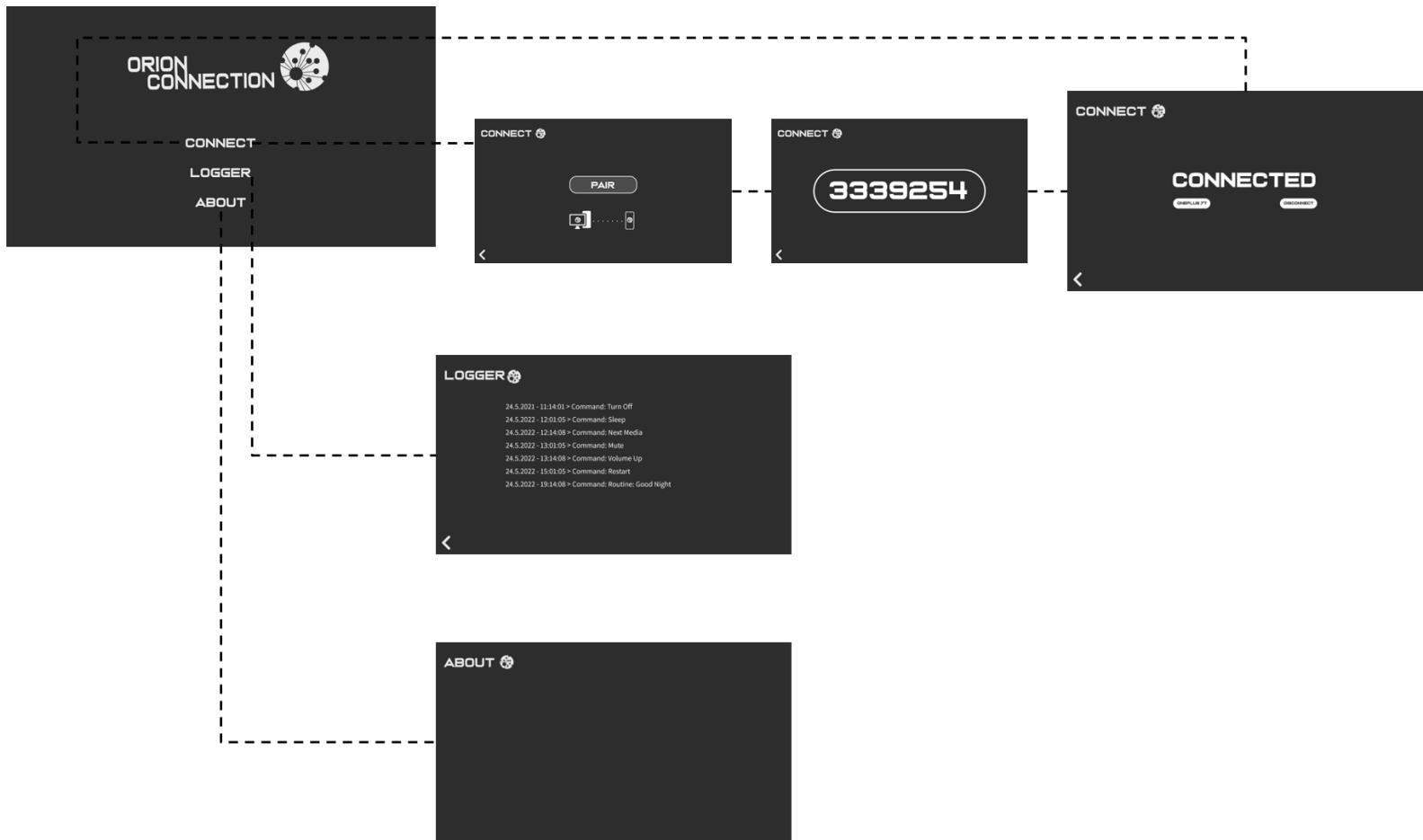
The screenshot shows a terminal window with the title "LOGGER". Inside, there is a list of log entries:

```
24.5.2021 - 11:14:01 > Command: Turn Off
24.5.2022 - 12:01:05 > Command: Sleep
24.5.2022 - 12:14:08 > Command: Next Media
24.5.2022 - 13:01:05 > Command: Mute
24.5.2022 - 13:14:08 > Command: Volume Up
24.5.2022 - 15:01:05 > Command: Restart
24.5.2022 - 19:14:08 > Command: Routine: Good Night
```

A large black rectangular redaction box covers the bottom portion of the terminal window, starting below the log entries and ending above the footer area. A small white arrow pointing left is located at the bottom-left corner of this redacted area.

מפרק המידע:

יכול בתוכו מידע כללי על המפתח ועל הפרויקט.

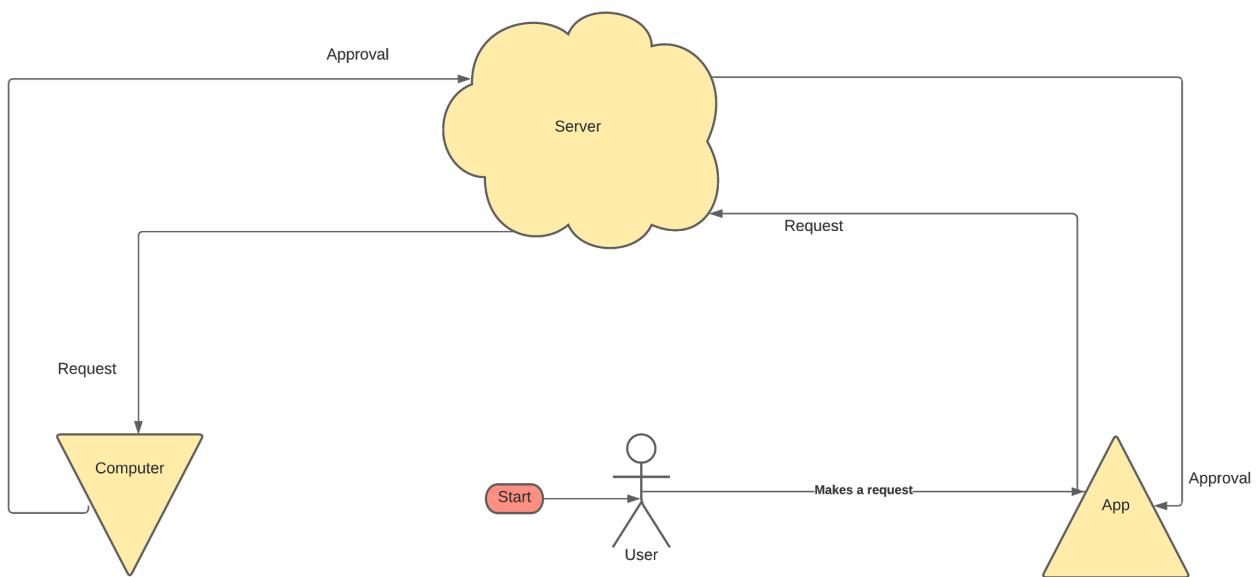
תרשים זרימה של מסכי תוכנת המחשב:

# Detailed Design

## תקשורות בין המכשירים

עקרון מפתח בפרויקט זה הוא התקשורת הציבורית בין טלפון למחשב. אין לגירום לטלפון לדעת שהוא שולט על מחשב מסוים. זאת על ידי בניית סרבר הפועל ב- IP ציבורי שיישמש כגשר בין השניים. הוא מקבל את ההודעות ויעבירו ליעד. כך אין זאת תקשורת Peer to Peer, זהו תקשורת בין סרבר מגשר - Bridge Server.

תרשים זרימה להעברת הודעות בין המכשירים:



התרשים מתרגם מעבר בין הודעות בין המכשירים -

- המשתמש מבצע פקודה דרך GUI האפליקציה בטלפון.

ההודעה עוברת לשרת.

- מהשרת ההודעה עוברת למחשב המתאים.

ואז ברגע ביצוע / קבלת הפעולה המחשב יחזיר הודעה אישור שתגיע בסופה של דבר דרך השרת אל מכשיר הסלולר.

## תהליך החיבור - Pair

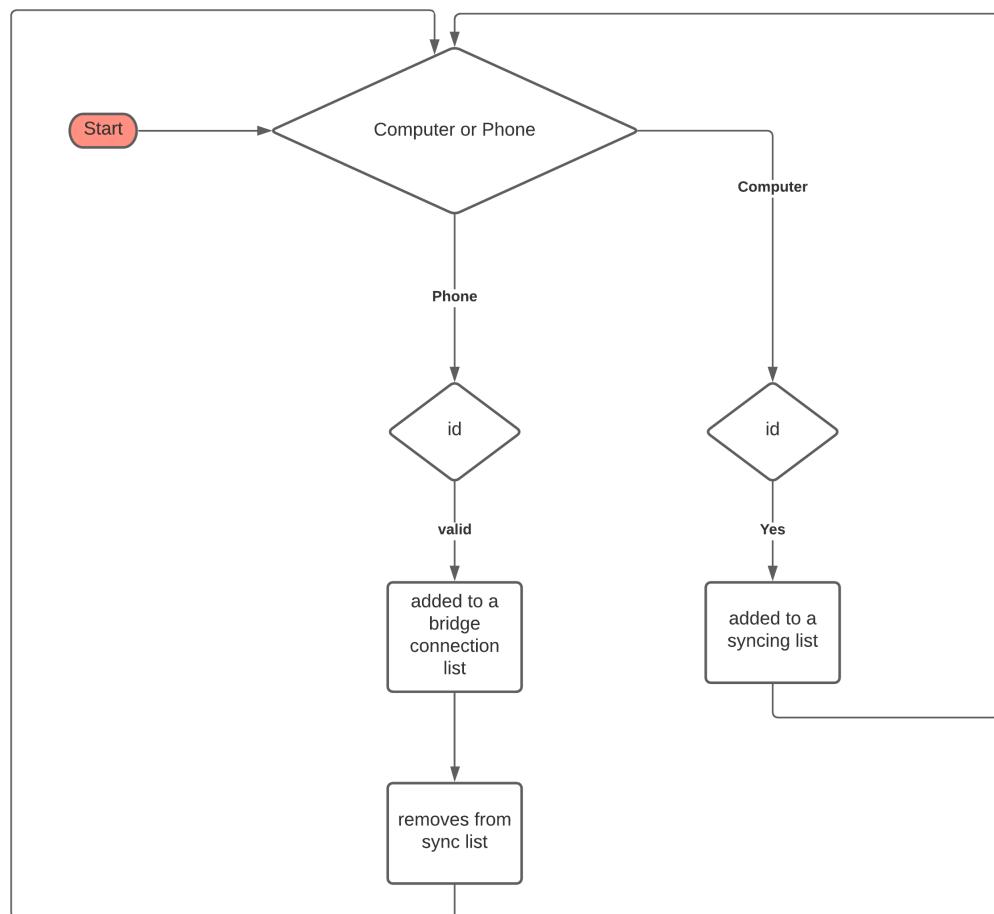
על מנת הגיעו במצב בו השרת יודע להבחן בקשר המתואם בין מחשב למכשיר סלולרי יש לבצע תהליך כל שהוא בו השרת ידוע להזין אובייקט קישור.

לכן תהליך ה-Pair יבוצע בצורה כזו של תוכנת מחשב יהיה קוד מיוחד משלו שיציג אותה מול הסרבר לדוגמה - 2541690.

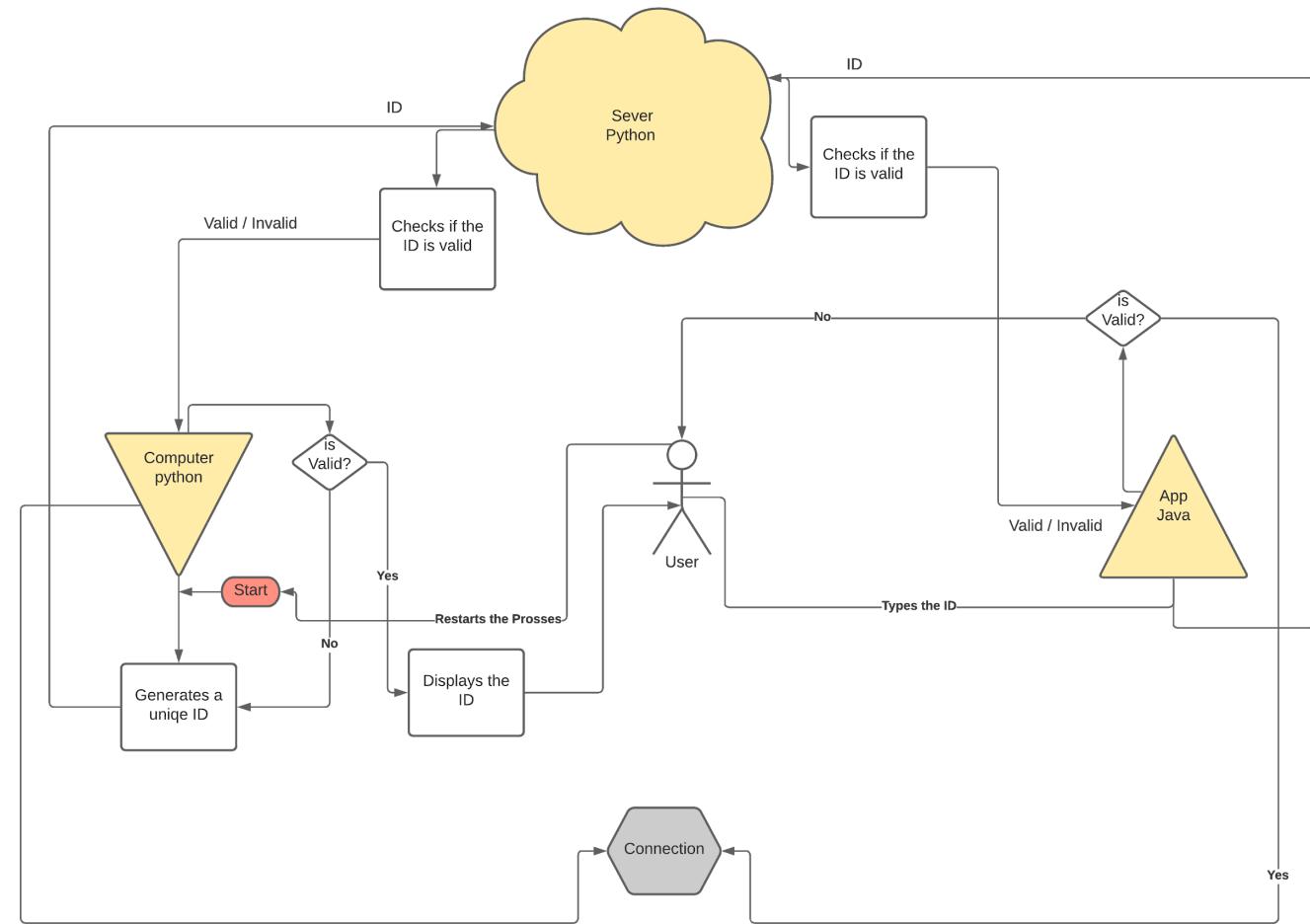
בעת תחילת תהליך הcoil דרך GUI תוכנת המחשב - התוכנה תציג את הקוד המיוחד בנוסף לסימנה ייחודיית רנדומלית לגמרי.

חיבור יעשה בצורה כזו שהיא משתמש יקליד את הקוד המיוחד המופיע באפליקציית הסלולר ולאחר מכן יקליד את הסימנה.

בצורה זו הסרבר יכנס לאובייקט מיוחד את הקישור בין השניים:



תהליך הcoil לכל צמד אינדיבידואלי יראה בצורה הזאת (כפי שהסביר קודם לכך) בדיאגרמה יותר מפורטת

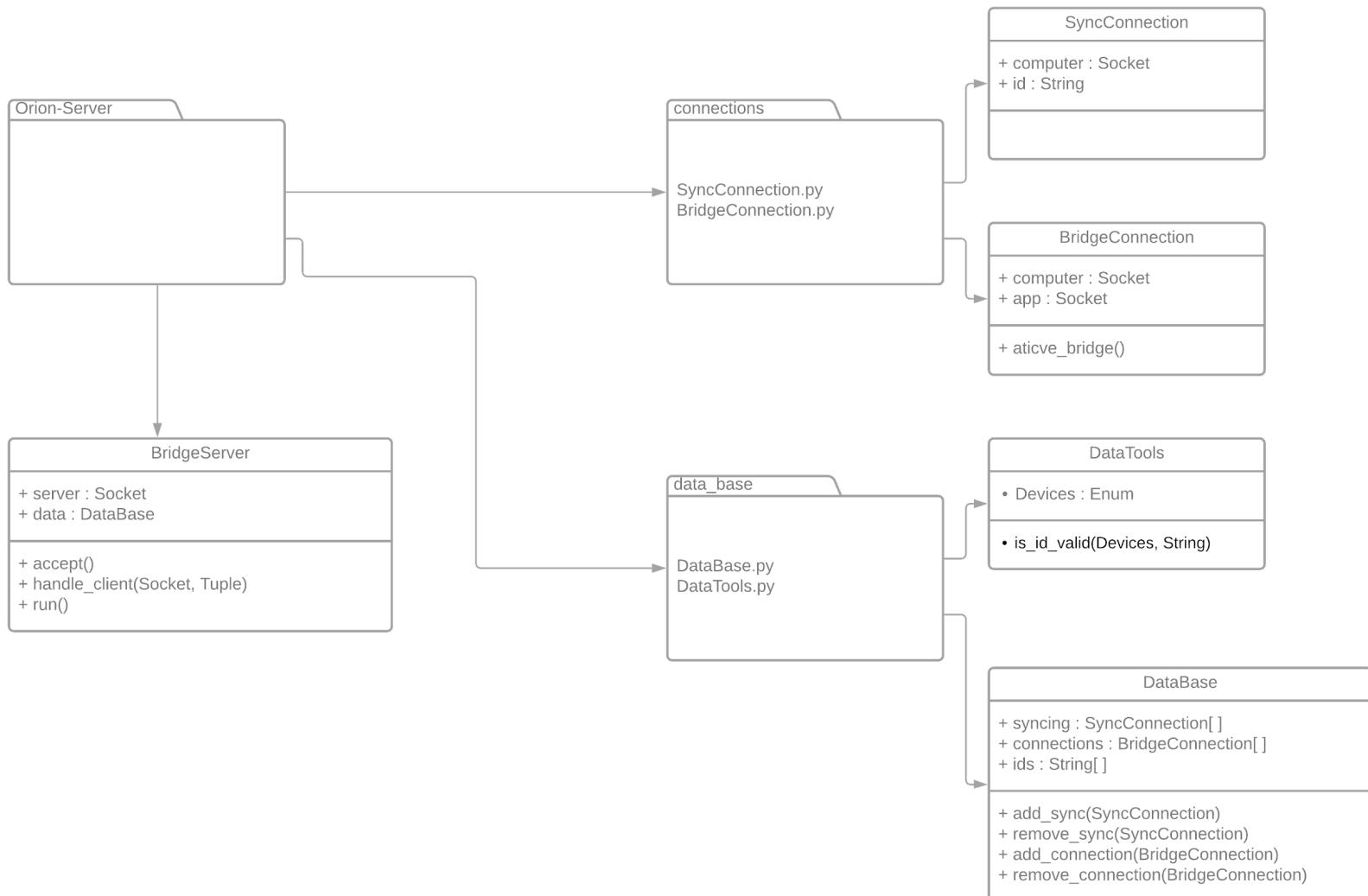


לפיעולות ממשיות בכל שלב בכoil בין המכניםים דרך השורת:

## מבנה הפרויקט

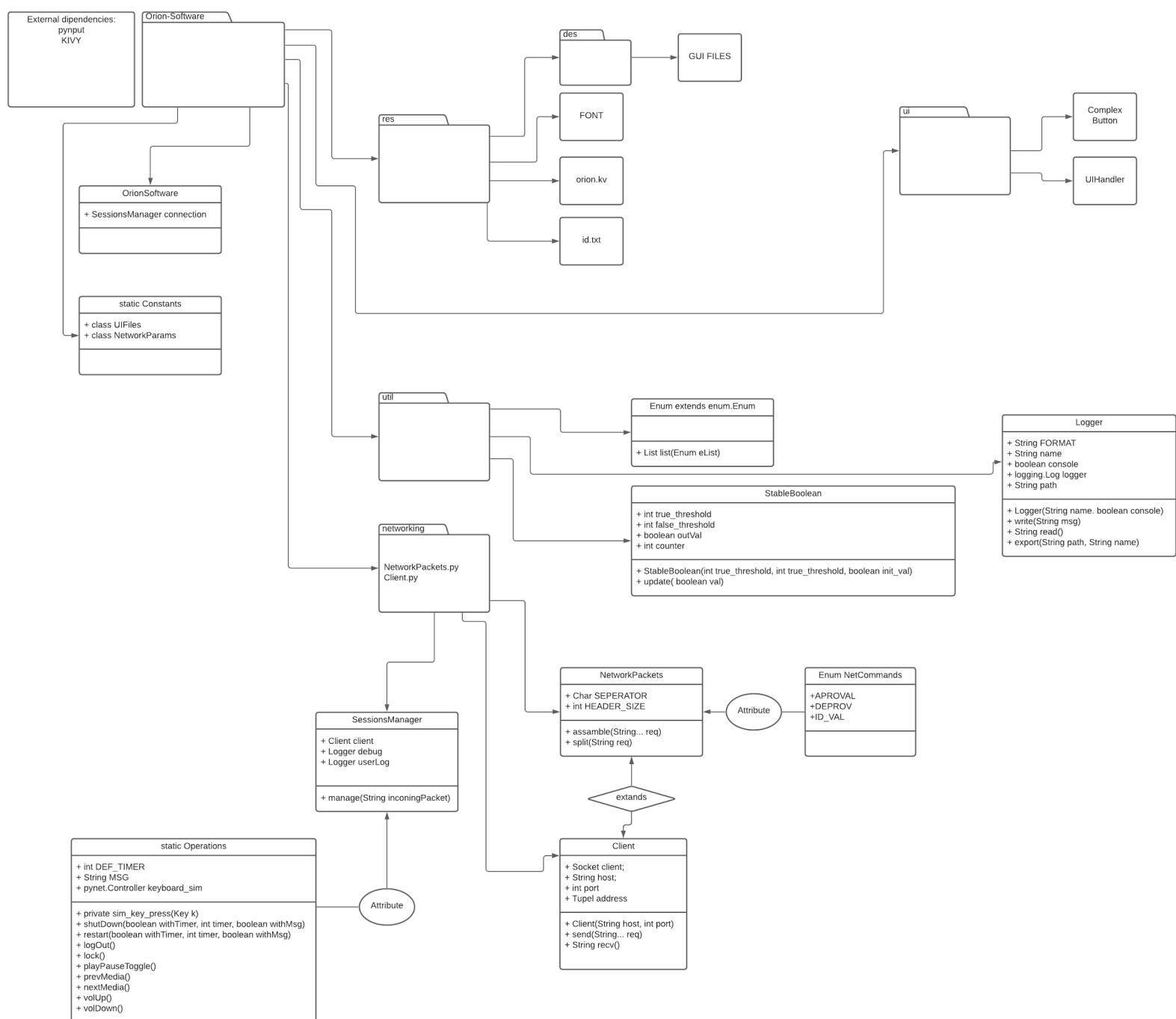
### השרות

השרות יכתב בשפת פיטון וירוץ על מחשב לינוקס בשרתים חיצוניים ברשות. מטרת השירות היא להיות גשר בין המכשורים ולנתב את ההודעות ביניהם בסדר ובכיוון הנכון. מצורפת דיאגרמת UML המתארת את מיקומי ושמות המחלקות, תכונות ופעולות בפרויקט השירות. החצים מסמלים כי מקור החץ מכיל את האובייקט (תיקייה או מחלקה) בסוף החץ.



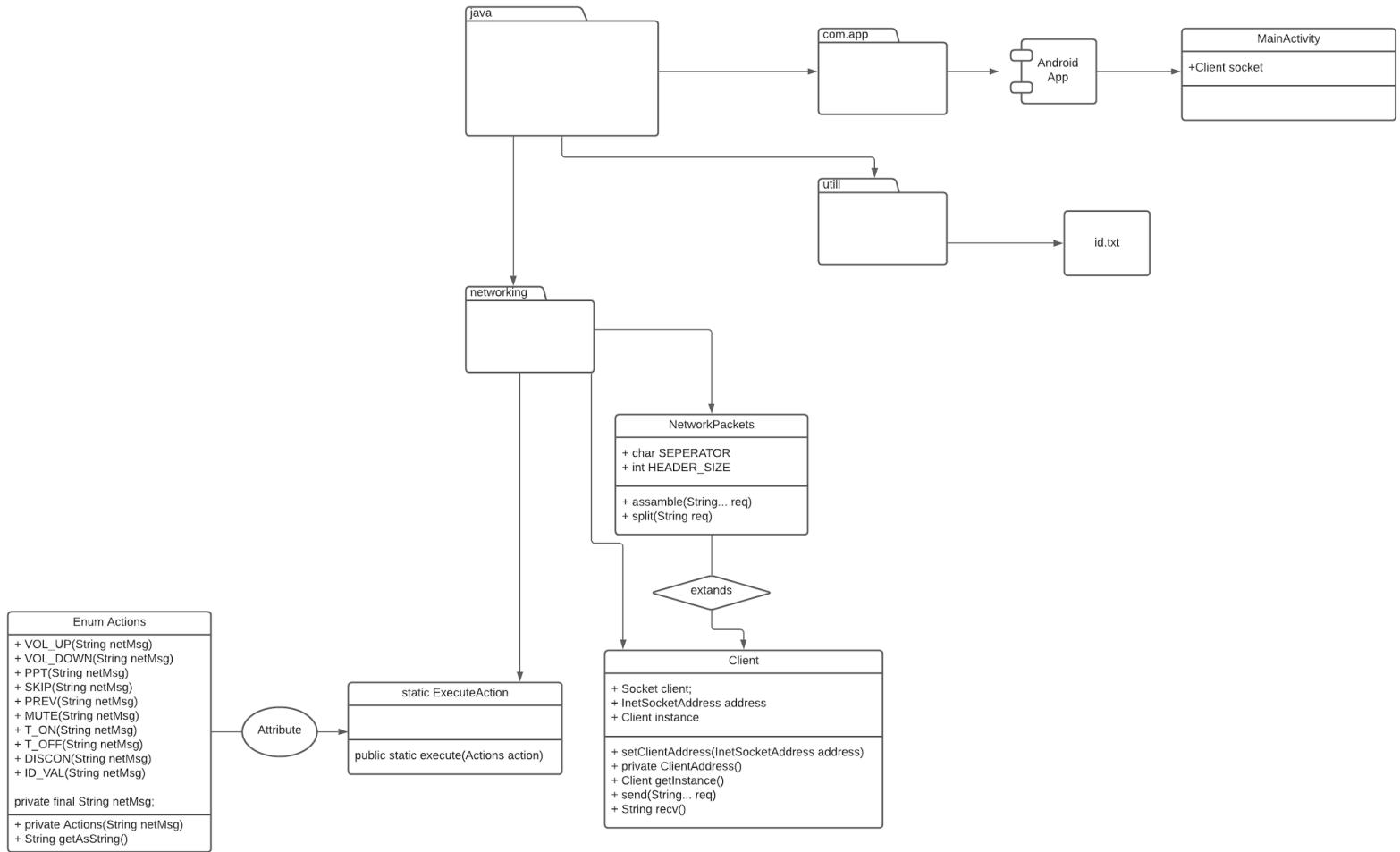
תוכנת המחשב

תוכנת המחשב תכתב בשפת פיטון + YAVI וירוץ על מחשב המשמש - ווינדוס. מטרת התוכנה היא לספק GUI מלא למשתמש בעת תחיליך הcoil והשליטה, לאחר מכן הקלעים היא תבצע את הפעולות השונות שהמשמש מבקש מורה. מצורפת דיאגרמת UML המתארת את מיקומי ושמות המחלקות, תכונות ופעולות בפרויקט תוכנת המחשב. החצים מסמלים כי מקור החץ מכיל את האובייקט (תיקייה או מחלקה) בסוף החץ.



## האפליקציה

האפליקציה כתובה בשפת XML + Java ותורץ על המכשיר הסלולרי של המשתמש - אנדרואיד. מטרת האפליקציה היא לספק ממשק GUI מלא לשילוטה על יכולות המחשב בצורה נוחה, מאחרו הkulium היא תעבור כל פקודה לשרת. מצורפת דיאגרמת UML המתארת את מיקומי ושמות המחלקות, תכונות ופעולות בפרויקט תוכנת המחשב. החצים מסמלים כי מקור החץ מכיל את האובייקט (תיקיה או מחלקה) בסוף החץ. יונש חצים מיוחדים המראים על קשר לוגי ותוכנתי - הורשה או תוכנה והכללה.



## פרוטוקולים

הקשר בין המכשירים יש מצריכה מגוון רחב של פקודות שייעברו דרך הרשת. מכיוון שליחת הודעה מסוימת עוברת דרך השרת ולאחר מכן למכשיר מטרן. אז, צריך שלוש סוגי פקודות כך שכל סוג מייצג את המכשיר שמשתמש (שרת, טלפון או מחשב):

מחשב	טלפון		שרת	סוג מכשיר
ID_VAL DISCON	ON OFF DICON ID_VAL SLEEP LCK RESTART	VOL_UP VOL_DOWN PPT SKIP PREV MUTE	VAL INVAL APROVDISCON	פקודות

## פקודות מהשרת

השרת מתעורר ושולח הודעות אך ורק בתהליך החיבור או סיום, אחרת הוא מתנהג כקשר שלא מתעורר בהודעות העוברות דרכו.

הסביר ומטרת	שם פקודה
נשלח בעת תחילת ההתחברות בהתאם לIFORMUL.	VAL/ INVAL
נשלח חוזה אל המכשיר שרצה להתנק על מנת לאשר את הפעולה.	APROVDISCON

## פקודות מחשב

המחשב לא שולח הודעות אלא מבצע את הפעולות. ההודעות היחידות שהוא מוכרכ לשלוח הן הודעות הקשורות בתחילת החיבור.

שם פקודה	הסבר ומטרה
ID_VAL	בתחילת חיבור לפי תחילת הכויל.
DISCON	נשלח כאשר יש רצון להתנתק

## פקודות מהטלפון

כפי שניתן לראות בטבלה לעיל, למושיר הסלולרי יש את המגוון הכי רחב של פקודות שהוא יכול לשנות. זאת מפני שלכל כפתור או פעולה שנייה לבצע ב-UIO יש לשולח הודעה אינטראקטית בהתאם. זאת בנוסף להודעות תחזוקתיות מאחוריו הקלים על מנת ליצור ולשמור על החיבור.

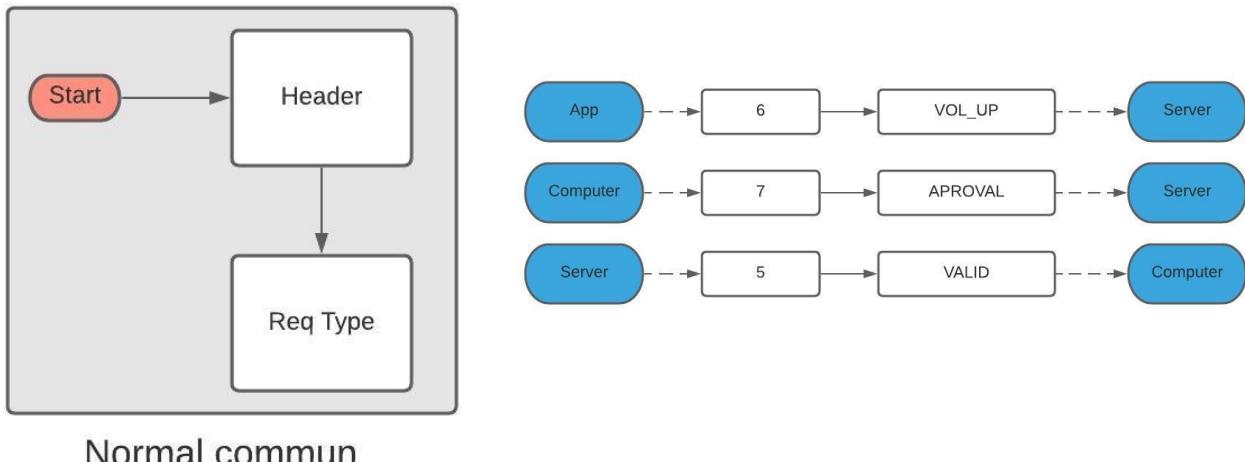
שם פקודה	הסבר ומטרה
VOL_UP / VOL_DOWN	נשלח בהתאם לחיצה על כפתורי שינוי הולמים.
PPT	נשלח בהתאם לחיצה על כפתור הטוגל של הפעל או עוצר. Play Pause Toggle.
SKIP / PREV	נשלח בהתאם לחיצה על כפתורי העברת שירים.
MUTE	נשלח בהתאם לחיצה על כפתור ההשתקה.
ON / OFF / RESTART / LCK / SLEEP	נשלח בהתאם לחיצה על אחד מכפתורי השליטה על מצב ההערות של המחשב.
DISCON	נשלח כאשר יש רצון להתנתק
ID_VAL	בתחילת חיבור לפי תחילת הכויל.

## מבנה הודעות

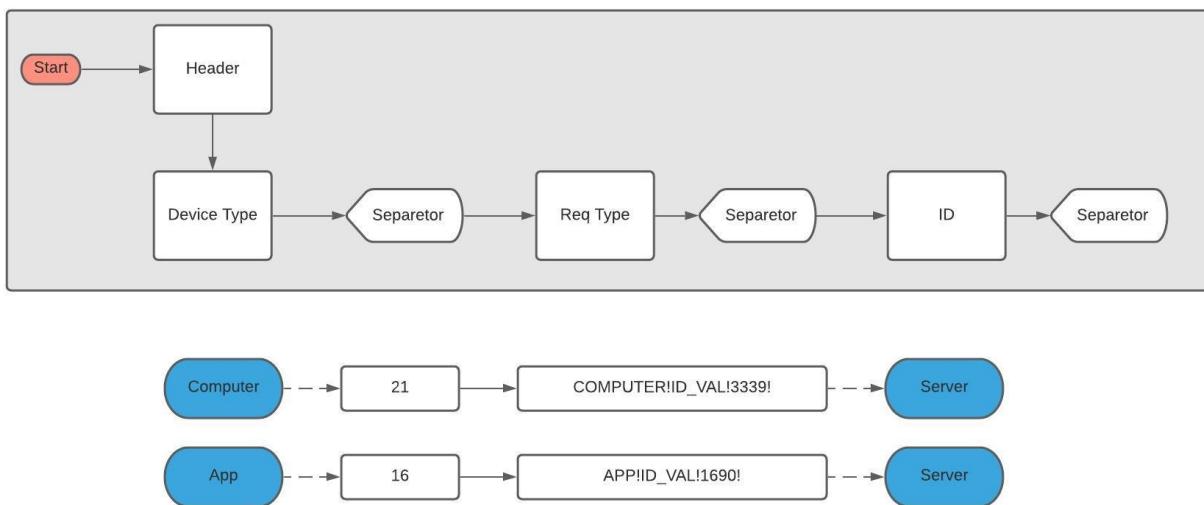
ישנם שני סוגי מבנים להודעות האינטראקטיביות:

- תקשורת רגילה - התקשרות השוטפת בין המכים
- תקשורת כoil - מהלך התקשרות הכלול את מהלך ההתחברות. השוני במבנה הודעות נובע ממידע נוסף ו שונה בתהליך החיבור.

### מבנה רגיל:



### מבנה לתחילה הסync'ו:



## רפלקציה

### למה בחרתי בפרויקט זה?

במהלך אותה תקופה הפכתי את החדר שלי לבית חכם. מנורות ורכזות לדימ הנשלטות מרוחק על ידי הטלפון והעזרה האישית של גוגל. הדבר האחרון שנשאר "טיפש" בחדר החכם שלי היה המחשב. רציתי את יכולת לשולט בו כמו כל מוצר אחר של בית חכם. לבסוף כשאנו הולך לשון, להדליק כשאנו חוזר הביתה, לעשות PLAY כשהמיטה רואה סרט.... התחלתי לחקור ולראות כל מיני אופציות, כל האפשרויות היו לא יציבות או מוגבלות בפתרונות שלhn. אז החלטתי ליצור בעצמי פיתרון אסתטי ונוח לבעה זו - להפוך כל מחשב נייד / נייח למוצר בית חכם.

### מה למדתי?

התחלתי את הפרויקט שבירשותי רוב הכלים הנדרשים לביצועו אך, היה חסר לי הידע בתכנות אפליקציות והידע התיאורטי של כל הפרויקט. התחלתי את הפרויקט במחקר עמוק על כל תחנת נשאי השוני - תקשורת, אנדרואיד, שליטה במחשב ועוד. כתה, אני יודע לתכנן וליצור אפליקציות, יש לי ידע רחב יותר באיך תקשורת בין רכיבים בבית חכם זו עובדת.

### תיכון זמני

את הפרויקט התחלתי ממש בחודשים הראשונים של כיתה יב. ידעת כי ברצוני לסיים אותו כמה שיותר מוקדם על מנת שאוכל לפניות את הזמן והלחץ ממנו לשאר הבגריות. ואכן כך היה, בתחילת פברואר כבר הפרויקט היה גמור 99% מהפתרונות שרציתי. ניצלתי את הקורונה והזמן הפנוי שהיא נתנה לעבוד במרץ על הפרויקט וכך להספק הכל. לאחר חודש פברוארעשיתי תיעודים, סידורים והוסףתי פיצ'רים אחרים על מנת ליצור פרויקט שהוא מושך בו במאות האחוזים.

### עיצוב UX / UI

מבחינת העיצוב עשית המון טווח עד שהגעתי למשהו שאני מרצה ממנו, למדתי על עיצוב נכון ומודרני ולמדתי להשתמש בכלי חדש ADOBE XD - כלי תעשייתי המשמש לבנית והדמית UX מלא.

## **כתיבת הפרויקט**

מכיוון שהפרויקט שלי מחולק ל-3 תתי פרויקט (סרבר, תוכנת מחשב ואפליקציה) התחלתי לכתוב כל אחד בנפרד (תוכנת מחשב -> סרבר -> אפליקציה).

השתדלתי לעבוד מסודר ועל פי פירוט ה-LML שהכנתי לפני תחילת כתיבת הקוד כך שתחליה ממש העתקתית את התבנית של ה-LML לקוד כתוב (הורשה תכונות ועוד).

החלק הכי מאתגר היה החיבור ביניהם במיוחד מצד האפליקציה, להעביר תקשורת אנדרואיד שתקרה מאחוריו הקלעים ללא כל הפרעה למשתמש וכן עוד אתגרים קטנים מצד אנדרואיד שהייתי עוד חדש לסבירבה זו.

הדברים שיותר אהבתי וכןלקח לי הרבה זמן הם עיצוב האפליקציה - מימוש עיצוב ה-UI ותוך קבצי ה-LXM שלה - כל פעם הופטוי ולמדתי משהו חדש ומגניב.

## **3 שנים בмагמת סייבר גבהים**

לאחר 3 שנים בмагמת סייבר גבהים אני יוצא עם 3 פרויקטים גדולים:

- [Assemble War](#)
- [Horizon Music](#)
- [Orion Connection](#)

על כל אחד מהפרויקטים עבדתי המון, למדתי והרחבתי את הידע והרמה התכניתית והאלגוריתמית שלי. הלמידה העצמית שהמגמה הציבה בפניו היא התרון המשמעותי שלא לעומת שאר מקצועות ביה"ס רגילים.

לאחר 3 שנים אני בהחלט גאה בהשגי יכול להגיד כי ההתקבלות שלי לגאמה היא חלק בלתי נפרד מהמגמה.

כל פרויקט גרם לי לשקו על תוך "FLOW" של עבודה שאף אחד לא הצליח להוציא אותה ממנה, הייתה חدور מטריה לפתור כל באג ולעקוּף כל מכשול שיעמוד בפני זאת למראות כי כל שנה לקחתי פרויקט קשה יותר מהנדרש.

**ORION  
CONNECTION**



## הוראות שימוש ומיתע כללי

Source Code: [App](#), [Server](#), [Software](#).

[Demo](#)

## תוכנת המחשב

**קובץ EXE יבוא בהמשך.**

### דרישות מערכת

- Windows 10
- על מנת להשתמש בפיצ'ר של הדלקה מרוחק - המחשב חייב להיות בעל חיבור אינטרנט קויי
- Python 3.7 ומעלה
- Kivy GUI Lib
- pip pyput
- pip hwmonitor

### הרצה

להגיע ולהריץ את- `OrionConnection.py` Orion-Connection-Software/src/ OrionConnection.py  
לחכמת שנייה ופתח חלון ה-UI של התוכנה.

### שימוש

בהנחה ששאר הרכיבים מוכנים.

#### חיבור ראשון

- להגיע למסך CONNECT וללחוץ על כפתור ה-PAIR
- להקליד במסך ה-PAIR במכשיר הטלפון את ה-ID שモופיע על המחשב

### MAGIC

- על מנת להפעיל את הגדרת MAGIC יש לכתוב את ה-PATH של התוכנה אותה רצים שהכפטור יפעיל.

### LOGGER

- הלוגר מספק מידע על כל הפעולות שבוצעו על המחשב עם פירוט זמן ותאריך מדויק. ניתן ליזא ולנקות את הקובץ דרך ההגדרות.

## אפליקציה

תגעה ל-PLAY STORE בהמשך.

### דרישות מערכת

- Android 9.0 •

### התקנה

להוריד דרך APK הנמצא [ב-GITHUB](#)

### שימוש

בהנחה ששאר הרכיבים מוכנים.

### חיבור ראשון

- ללחוץ על כפתור ה+ הנמצא מצד ימני העליון של המספר הראשי
- להקליד את ה-ID הנמצא על מסך המחשב תחת המיקום המתאים

### MAGIC

- על מנת לשנות את הגדרת MAGIC יש ללחט להגדרות ולבחר את אופציית השימוש הרצiosa.

### GROUP

- ניתן ליצור קבוצות המורכבות ממספר קישורים ייחודיים ולקבוצות - במספר GROUP דרך התפריט הראשי.

### ROUTINES

- ניתן ליצור תזמנונים לקישורים ייחודיים ולקבוצות - במספר ROUTINES דרך התפריט הראשי.