

Dorms Access Manager

מגישים:

חמדאן אגבאריה

עודאי שמאלה

באסל אבופול






פרויקט Dorms Access Manager הינו מנגנון בקרה וניהול לתהליך הכניסה למעונות הסטודנטים, כאשר המטרה העיקרית שלו היא ליעל תהליך זה ולהפוך אותו לאוטומטי ככל האפשר.

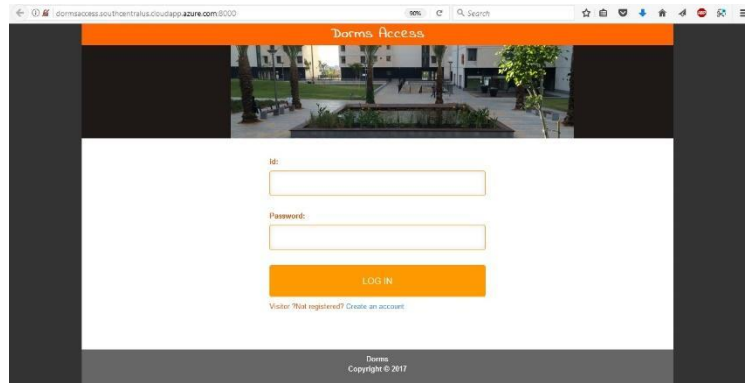
● מרכיבי הפרויקט:



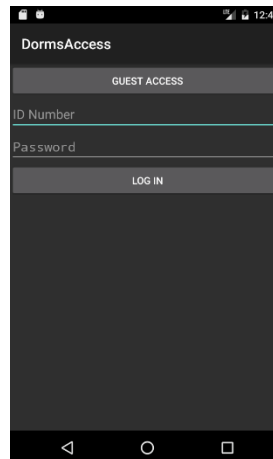
(1) מכשיר Raspberry pi 2.

(2) מבונה וירטואלית ב Azure שמריצה שרת ואתר.

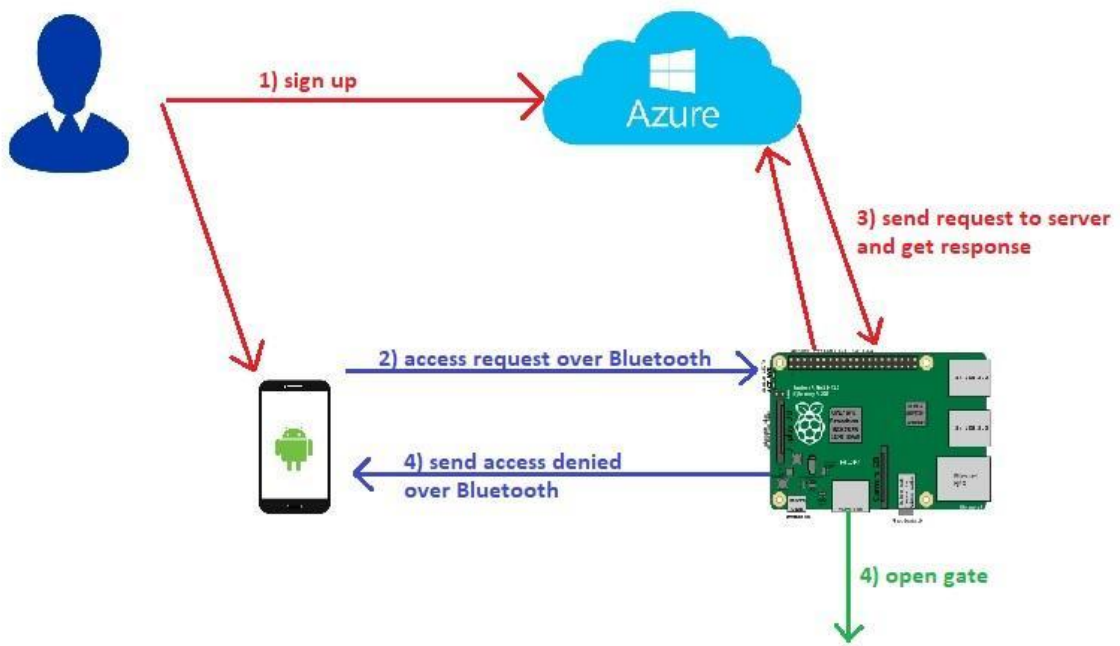
All resources ALL SUBSCRIPTIONS	
	dormsServer251 Network interface
	dormsServer Virtual machine
	dormsServer-ip Public IP address
	dormsaccessdiag335 Storage account
	dormsAccess-vnet Virtual network
	dormsServer_OsDisk_1_071ee3... Disk
	dormsServer-nsg Network security group



(3) אפליקצית Android.



• איך עובד שירות DORMS Access?



- המשתמש נרשם לשירות דרך האתר או האפליקציה.

DORMS Access Manager

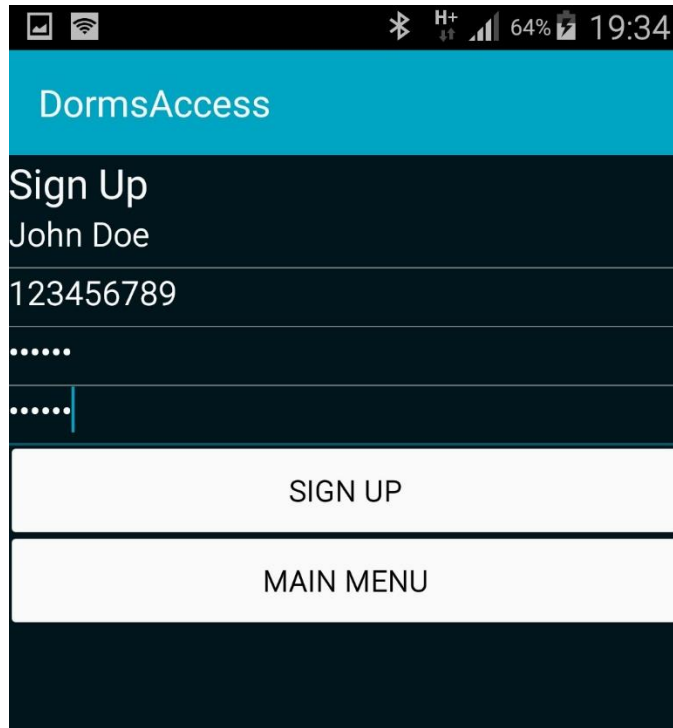
Id: 123456789

Name: John Doe

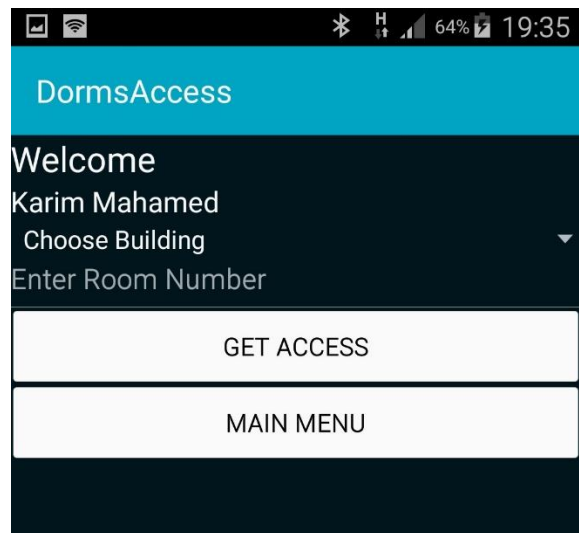
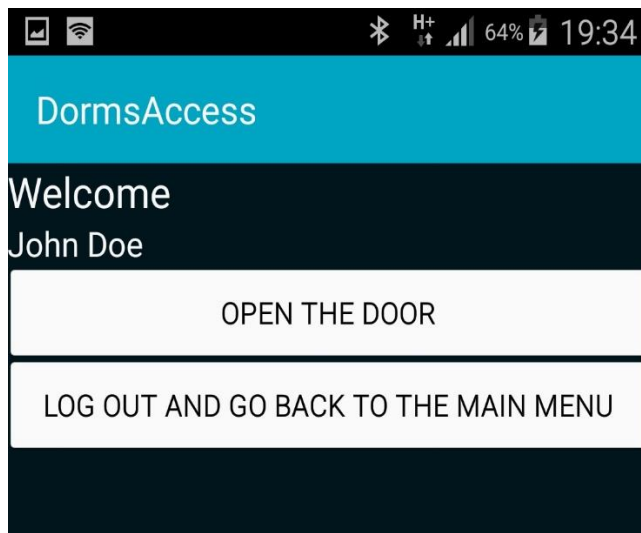
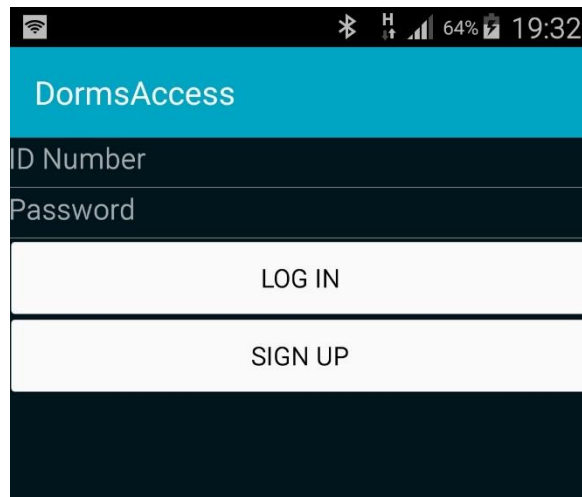
Password:

SIGN UP

DORMS Copyright © 2017



- כשהוא מגיע לכניסת המעונות הוא מפעיל את האפליקציה ושולח בקשת כניסה מתאימה ל Raspberry pi.



- ה Raspberry pi קולט בקשה זו ושולח אותה לענן על מנת לקבל אישור לפתוח את השער.
- השרת מקבל את הבקשה מה Raspberry pi ובודק אם היא חוקית או לא ומחזיר תשובה מתאימה ל Raspberry pi.

בודקים אם המשתמש נמצא ברשימת הדיירים

Dorms Access Manager

Residents | Add New Resident | Visitors | Access | Blacklist | Logs | Logged as: admin1 | Log Out

Residents List

Audai Shmalih , H0418 , 308010024	🗑️
Basel Abo Foul , E0902 , 123456781	🗑️
Karim Mahamed , H113 , 203928342	🗑️
John Doe , C1402 , 123456789	🗑️

SAVE

או אם הוא אינו דייר אזי בודקים אם יש לו הרשאה לבקר את הדייר והדירה שנשלחו בבקשה שלו

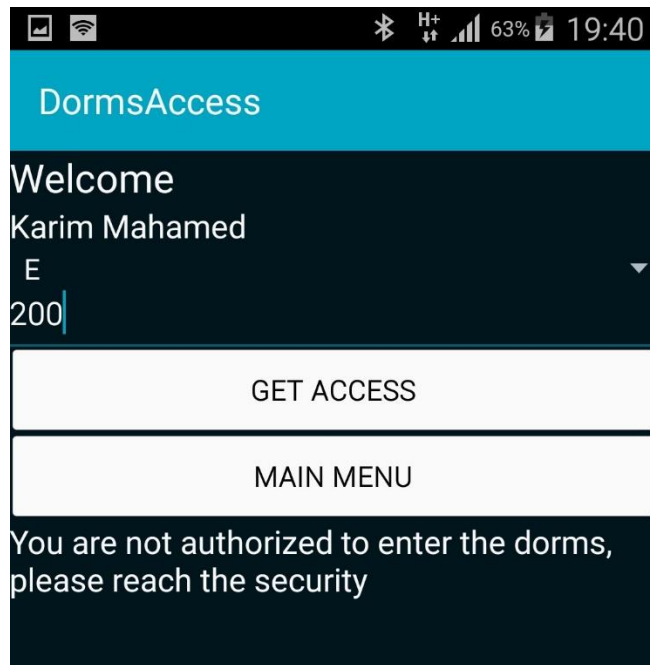
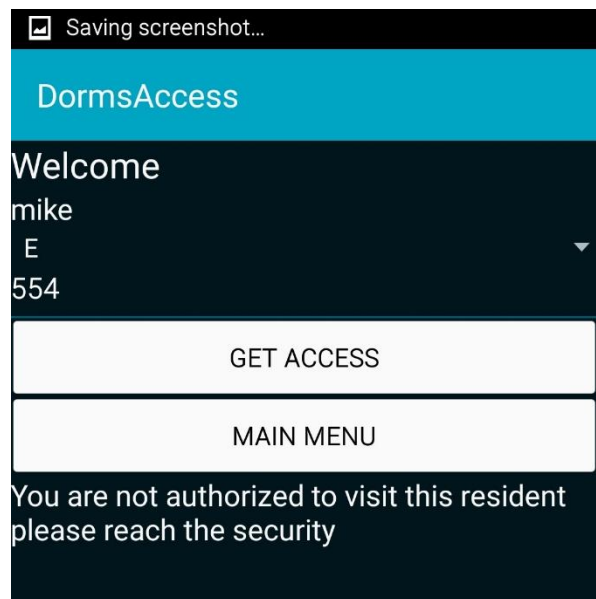
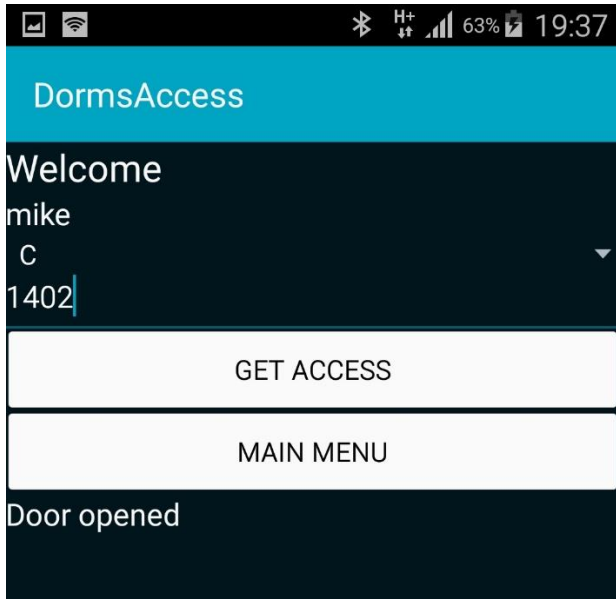
Dorms Access Manager

Residents | Add New Resident | Visitors | **Access** | Blacklist | Logs | Logged as: admin1 | Log Out

Access

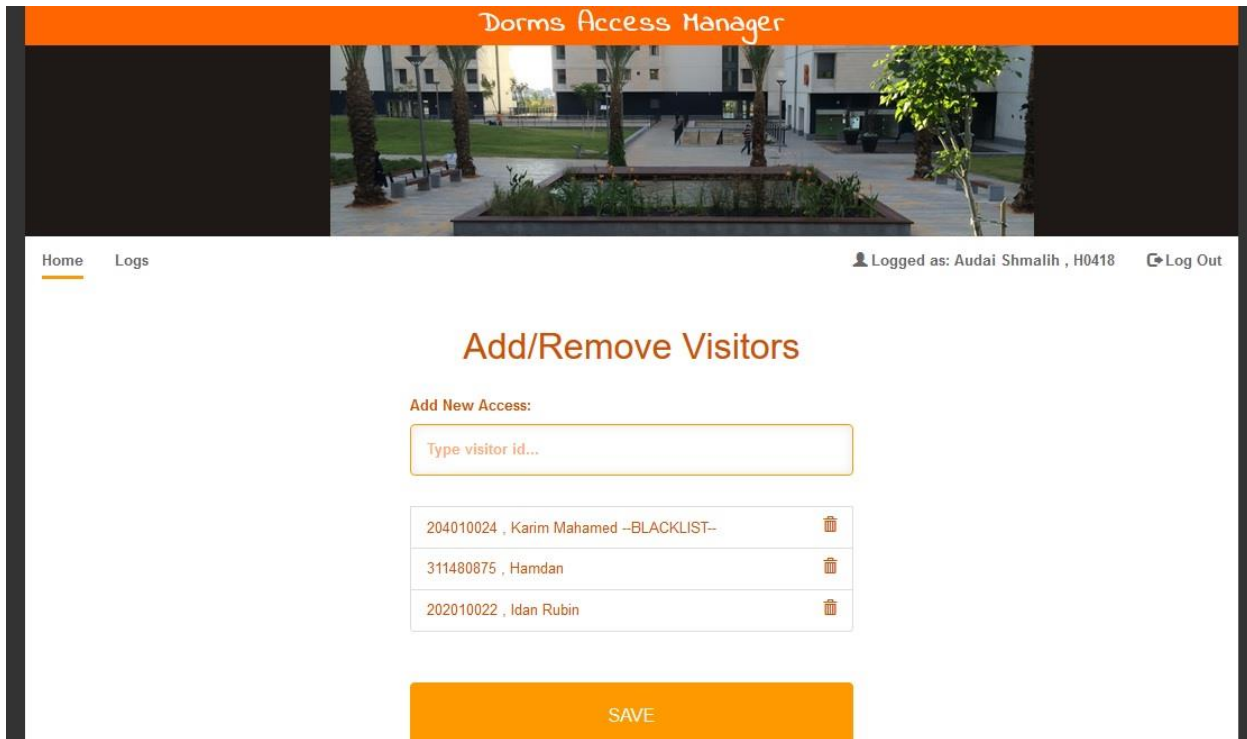
202010022 , Idan Rubin , H0418	🗑️
204010024 , Karim Mahamed , H0418	🗑️
204010024 , Karim Mahamed , C1402	🗑️
202010022 , Idan Rubin , C1402	🗑️
101010211 , Moshe Barkan , C1402	🗑️
202010022 , Idan Rubin , E0902	🗑️
101010011 , Johnson walker , E0902	🗑️
311480875 , Hamdan , H0418	🗑️

- ה Raspberry pi קולט את תשובת השרת ופותח את השער או מחזיר הודעת איסור כניסה למשתמש.

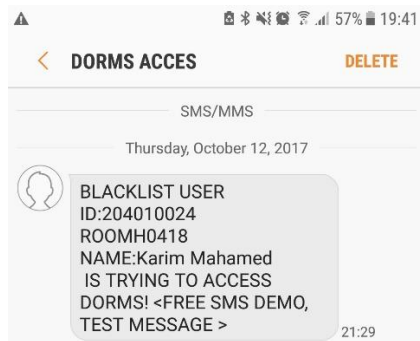


● בקרת הכניסה:

לכל משתמש שהוא דייר במעונות קיימת רשימה של מבקרים מורשים. רשימה זו מכילה את כל המשתמשים שיכולים לבקר את הדייר, והוא יכול לערוך רשימה זו בכל עת באתר או באפליקציה.



בכל פעם שמשתמש מנסה לבקר דייר אנו קודם כל בודקים אם השם שלו נמצא ברשימה השחורה של משתמשים שאסור שייכנסו למעונות, אם השם שלו מופיע שם אז מחזירים לו הודעת Access Denied מצאימה, ושולחים הודעת sms לביטחון המעונות כדי להודיע להם על המקרה.



אם המשתמש עובר את בדיקת ה blacklist בהצלחה אזי אנו בודקים אם השם שלו קיים ברשימת המבקרים המורשים, אם כן אז מאפשרים לו להיכנס ושולחים הודעת sms לדייר כדי להודיע לו על הביקור, אחרת, אנו שולחים למשתמש הודעת Access denied מתאימה.



- גישת Admin: כדי לעזור בניהול וההשגחה על תהליך הביקור אנו מאפשרים גישה עם הרשאות מיוחדות באתר השירות. משתמש שהוא Admin יכול:

- למחוק ולהוסיף משתמשים

Visitors

Residents List

204010024 , Karim Mahamed



Audai Shmalih , H0418 , 308010024



Residents Add New Resident Visitors Access Blacklist Logs

Logged as: admin1 Log Out

Resident Id:

Name:

Room:

Password:

ADD NEW RESIDENT

- לשנות Access של משתמשים

Access

202010022 , Idan Rubin , H0418



- לערוך את רשימת ה blacklist

Dorms Access Manager

Residents Add New Resident Visitors Access **Blacklist** Logs

Logged as: admin1 Log Out

Blacklist

Add User To Blacklist

Type user id...

Karim Mahamed , visitor , 204010024



Karim Mahamed , resident , 203928342



SAVE

- לראות היסטוריית כניסות וביקורים מלאה, ולבצע בה חיפוש לפי תעודת זהות, חדר ותאריך

Dorms Access Manager

Residents Add New Resident Visitors Access **Blacklist** **Logs**

Logged as: admin1 Log Out

LOGS

Search In Logs:

None

Type id/room/date...

2017-10-14 19:10:42 | C1402 | 123456789 | John Doe

2017-10-14 19:44:36 | H0418 | 311480875 | Hamdan

2017-10-14 20:08:16 | C1402 | 123456789 | John Doe

2017-10-14 19:06:50 | C1402 | 123456789 | John Doe

• מידע נוסף על הרכיבים ותהליך הפיתוח:

1) Raspberry pi:

המכשיר מריץ שרת Bluetooth בפרוטוקול RFCOMM שמאזין למשתמשים שיתחברו אליו דרך האפליקציה, וברגע שחיבור מתקבל הוא מתחיל לתקשר עם המשתמש. פרוטוקול RFCOMM נבחר מכיוון שהוא קל לשימוש ומאפשר שליחה פשוטה של Strings מעל Bluetooth, וזה אפשר לנו ליצור פרוטוקול תקשורת מבוסס Strings בין ה Raspberry pi ואפליקצית האנדרויד, כאשר כל בקשה מהאפליקציה מומרת לצורת String ונשלחת דרך ה Bluetooth. בקשות הכניסה והביקור עוברות רק דרך Bluetooth אל ה Raspberry pi כדי לוודא שהמשתמש אכן בסביבת המכשיר לפני שפותחים את השער, וברגע שבקשה מתקבלת המכשיר שולח בקשת Http מתאימה לשרת, ואז לפי תשובת השרת הוא שולח הודעה מתאימה לאפליקציה. לאחר סיום התקשורת עם המשתמש הנוכחי ה Raspberry Pi חוזר לשלב ההאזנה.

הקוד נכתב בשפת C# עבור מערכת הפעלה Windows 10 Core, ותהליך הפיתוח כלל כמה אתגרים וקשיים כאשר רוב ההסברים של מייקרוסופט מסתכמים בדוגמאות קוד ללא הסבר או הדרכה.

אחת הבעיות המרכזיות הייתה עם קוד שרת ה Bluetooth, כאשר קוד הדוגמה של מייקרוסופט רץ על מחשב רגיל, אך אינו עובד כאשר מנסים לעשות deploy ולהריץ אותו על ה raspberry pi. בעיה זו נפתרה לאחר ניסוי קומבינציות שונות של הרשאות גישה של הקוד ב Raspberry pi עד שמצאנו את הקומבינציה המתאימה.

בעיה נוספת עם שרת ה Bluetooth הייתה שהשרת נכבה לחלוטין לאחר שהוא מסיים את התקשורת עם client כלשהוא, ולא מאפשר ולחזור לשלב ההאזנה באופן מיידי, ואם רוצים להאזין עד לחיבוא ה client הבא אנו צריכים להריץ את קוד אתחול השרת שוב. לבעיה זו לא מצאנו פיתרון אלגנטי, בעיקר בגלל שקוד השרת רץ ב thread נפרד משאר הקוד ולכן כל קריאה לפונקציה מתוך קוד השרת עלולה הייתה להיתקל בשגיאות context או scheduling. קוד הדוגמה של מייקרוסופט לשרת RFCOMM עבד באופן דומה ולכן הייתה לו את אותה הבעיה. בסופו של דבר, בעיה זו נפתרה ע"י שימוש במשתנים אטומיים לסימון סיום ריצת חוט השרת, כאשר קיימת לולאה אינסופית ב main thread שבודקת משתנים אלה כדי לקבוע מתי להפעיל את קוד אתחול השרת מחדש.

2) Azure:

בענן יש Web application שמורכבת מ:

- בסיס נתונים Mysql שמכיל כל הטבלאות שאנחנו את צריכים.
- שרת של האפליקציה שמטפל בבקשות שמקבל מצד הלקוח של ה **Web application** וגם מטפל בבקשות (**Rest api**) שמקבל מאפליקציית האנדרויד וגם מה Raspberry pi, צד השרת נכתב בשפת **Python** שהשתמשנו ב **Django web framework** בחרתי להשתמש ב Framework הזה כי ראיתי שהוא ממש נוח לכתוב בפיתוח ואני שולט בשפה הזאת וגם לחלק ולנהל את הקוד בצורה נוחה ולעבד את הנתונים ולשמור/לשלוף/למחוק מבסיס הנתונים בצורה קלה.
- צד הלקוח: הקוד ה־ HTML, CSS ו־ JavaScript.

אתגרים ובעיות: בהתחלה ניסיתי להעלות את השרת דרך שימוש ב UI של Azure כ Web application למרות שזה נראה קל בהתחלה התברר לי שזה די מסובך להעלות שרת שנכתב ב Python ושימוש ב Django וגם יש כמות קוד גדולה בשרת להעלות את זה בצורה כזאת כקופסה שאתה לא ממש מבין איך עובד יצר תחושה לא טובה, אחרי עבודה של כמה ימים ללא הצלחה, החלטתי לעשות את זה בדרך אחרת ופתחתי Virtual Machine עם מערכת הפעלה Ubuntu, ומשם יש לי את Ip של המכונה ויכול להתחבר אליה SSH דרך ה MobaXterm ואז עשיתי כל ההתקנות שאני צריך על המכונה וזה עבד.

בסיס הנתונים : בהתחלה פתחתי דרך ב UI של Azure אבל אחרי שפתחתי מכונה וירטואלית החלטתי לעשות את זה אחרת שבמקום לשלם על זה בנוסף , אני יכול לבנות אותו על המכונה אז אני לא אצטרך לשלם וגם יותר יעיל ונוח להתחבר אליו מהשרת.

פיצ'ר נוסף- שליחת SMS מהשרת :

אחרי שסיימנו את בסיס הפרויקט, חשבנו גם להכניס עוד פיצ'רים נוספים, ואז עלה לנו את הרעיון שליחת SMS, חקרתי באינטרנט איך אפשר לעשות דבר כזה מצאתי כמה ספריות כמו TWILIO אבל דורש עוד עבודה ובתיבת קוד התחלתי עם זה, אבל אחר כך גיליתי שירות אחר של Vonage שנקרא Nexmo שיותר פשוט, אתה ממלא הכל דרך האתר שלהם ואז אתה יכול לעשות בקשת POST עם הטקסט ולאן והשירות שלהם מפרסר את את זה ושולח אותו ליעד.

(3) אפליקציה:

השתמשתי ב xamarin visual studio לבניית האפליקציה והיא נכתבה בשפת #C.
באופן כללי תהליך בניית האפליקציה היה חלק, ולא היו בעיות או קשיים מיוחדים בתהליך הפיתוח. ה visual studio מספק את כל הכלים הדרושים לבניית אפליקציה קטנה, והוא מאוד קל לשימוש.
פירוט על אופן השימוש באפליקציה נמצא בתיעוד למעלה.