**עבודת גמר תכנון ותכנות מערכות 883599**



תאריך: 10.05.2020

**תוכן עניינים**

מבוא .............................................................................................................................3

מדריך למשתמש..............................................................................................................4

תפעול המערכת..........................................................................................................5-13

תרשים זרימה בין המסכים...............................................................................................14

הרשאות ודרישות מיוחדות...............................................................................................15

מדריך למתכנת..................................................................................................................16-17

תרשים UML............................................................................................................................18

Activities................................................................................................................19-21  
Service & Broadcast......................................................................................................22  
טבלת מחלקות.........................................................................................................23-25  
שיקולי תוכן..................................................................................................................26  
Resources...............................................................................................................27-30  
תפריט..........................................................................................................................31  
Drawables..............................................................................................................32-33  
Raw.............................................................................................................................34  
טבלת קבצים.................................................................................................................35

סיכום...........................................................................................................................36  
נספחים........................................................................................................................37  
ביבליוגרפיה..................................................................................................................38

**מבוא**

שם האפליקציה : QuizTime

האפליקציה QuizTime היא אפליקציית טריוויה. באפליקציה קיימות שלוש קטגוריות שונות : חלל, גיאוגרפיה והיסטוריה, כאשר תחת כל קטגוריה יש חמש שאלות שונות. השאלות דורשות ידע כללי בנושאים ללא העמקה בהם, כלומר אלה עובדות בסיסיות שכל בן אדם אמור לדעת.  
כמו כן, אפליקציה זו מיועדת לילדים ומבוגרים כאחד ואין לה מגבלת גיל.

בכדי להתחיל את המשחק על המשתמש להירשם לאפליקציה או להתחבר עם משתמש קיים, ומשם המשתמש מועבר לעמוד של הקטגוריות. בעמוד של הקטגוריות על המשתמש לבחור את הנושא שהוא רוצה להיבחן בו. לפני תחילת השאלון מופיע עמוד עם הסבר על הנושא שבו בחר וכפתור חזרה לעמוד הקטגוריות במקרה ושינה את דעתו לפני תחילת השאלון.

כאשר מתחיל השאלון המשתמש יכול לראות טיימר בחלקו העליון של המסך. טיימר זה הוא סוג של ספירה לאחור מאז שהמשתמש התחיל את השאלון. הספירה לאחור היא משתי דקות עד לאפס, כלומר הזמן המקסימלי שיש למשתמש בכדי לענות על השאלון כולו הוא שתי דקות. בנוסף, המשתמש יכול בסוף השאלון לראות את כמות התשובות הנכונות שענה מתוך סך השאלות הקיימות. מעבר לכך באמצע המשחק עצמו מוצגת בפני המשתמש הודעה האומרת האם הוא טעה או צדק. במקרה שטעה, מוצגת התשובה הנכונה וכפתור המשך לשאלה הבאה. במקרה שצדק מוצג משפט מעודד וכפתור המשך לשאלה הבאה.   
המשתמש לא יכול לעבור לשאלה הבאה בלי לענות על השאלה הקודמת.

כמו כן, ישנה אפשרות באפליקציה על הדלקת מוזיקת רקע במהלך הזמן שבו שוהה המשתמש הרשום במשחק, כלומר המשתמש הרשום רשאי לבחור האם ברצונו לשחק עם המוזיקה ברקע או בלעדיה. אפשרות זו נמצאת בתפריט נשלף במסך של נושאי הטריוויה.  
בנוסף לאפשרות של הדלקת מוזיקת רקע ישנה התראה המודיעה על שינויים במצב הטיסה בטלפון.  
אם המשתמש הדליק או כיבה את מצב הטיסה מופיעה למטה הודעה האומרת כי מצב טיסה השתנה.

השאלות המוצגות הן מילוליות, בצורת שאלון אמריקאי. כלומר מוצגת בפני המשתמש שאלה כתובה בטקסט, ארבע אפשרויות תשובה כאשר רק תשובה אחת היא הנכונה וכפתור של שליחת שאלה.

בחרתי לבנות אפליקציה דווקא מסוג טריוויה/שאלונים משום שאני מאוד אוהבת משחקים בסגנון דומה. אני חושבת שאפליקציות כאלה מלמדות הרבה ומשפרות את הידע הכללי של המשתמשים בהן. כמו כן, בימינו אנשים מבלים הרבה מאוד זמן במכשירים הסלולריים שלהם ולרוב זמן בילוי זה חסר כל תועלת ומתבטא בגלישה בעמוד אינסטגרם, פייסבוק וכדומה, כך ששעות על גבי שעות מתבזבזות על פעילות סתמית. לעומת זאת אפליקציות מסוג טריוויה, חידות וכדומה הן אפליקציות שמפתחות את הדמיון, חשיבה מחוץ לקופסה וכמובן את הידע הכללי. אני חושבת שאנשים יסכימו איתי שעדיף "לבזבז" את הזמן הפנוי שלנו על העשרת הידע הכללי שלנו ולא על רשתות חברתיות שגוזלות זמן רב ללא שום תרומה אמיתית ומוחשית.

**מדריך למשתמש**

**מטרת המערכת :**

מטרת האפליקציה היא לבנות משחק טריוויה מהנה וכיפי אך באותו הזמן גם מלמד ומרחיב את הידע הכללי של המשתמשים, לשם העברת זמן פנוי ביעילות.

**איך משחקים ?**

לפני תחילת המשחק על המשתמש להירשם לאפליקציה, תנאי זה הוא חובה, ומשם הוא יעובר למסך הקטגוריות. במסך הקטגוריות בוחרים באחד מהנושאים הקיימים שמעניינים את המשתמש או שברצונו ללמוד עליהם יותר. לפני תחילת השאלון יהיה הסבר קצר על הנושא שנבחר ומשם כאשר מתחיל השאלון למשתמש יש **שתי דקות** **בכדי לענות על כל השאלון.** לכל שאלה יש ארבע אפשרויות בחירה כאשר **רק אחת מהאפשרויות** **מהווה תשובה נכונה** לשאלה המוצגת. בסוף השאלון המשתמש יוכל לראות בכמה שאלות הוא צדק מתוך סך השאלות הקיימות בנושא שנבחר ומשם הוא יוחזר למסך הקטגוריות.

**יכולות המשחק :**

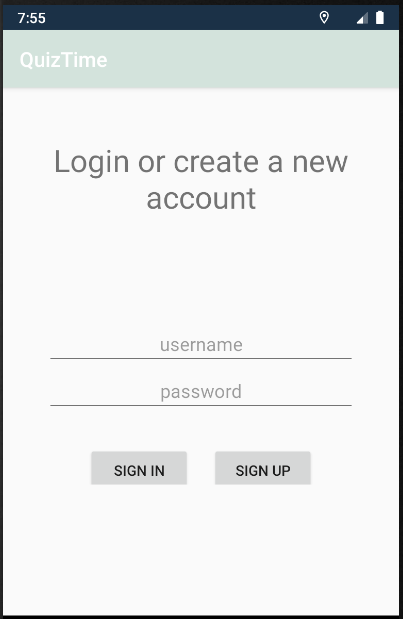
במשחק קיים מגוון של אפשרויות בכדי להפוך את זמן הבילוי של המשתמש באפליקציה לכמה שיותר מהנה וכייפי.

* אפשרות להדלקת וכיבוי מוזיקת רקע במסך הקטגוריות, בתפריט הנשלף מלמעלה.
* אפשרות להרשמה והתחברות.
* אפשרות של צפייה בכמות התשובות הנכונות מתוך סך השאלות בסוף כל שאלון.
* אפשרות לבחירת קטגוריה בה המשתמש ירצה להיבחן.
* אפשרות של התראה במידה ומצב טיסה השתנה.
* אפשרות של הודעה על האם התשובה שנשלחה היא נכונה או לא נכונה בזמן השאלון עצמו.
* אפשרות של צפייה בתשובה נכונה לשאלה בזמן השאלון במידה והתשובה שניתנה היא שגויה.
* אפשרות לראות בחלקו העליון של המסך בזמן השאלון מה היא כמות הזמן שנשארה מתוך שתי הדקות לסוף השאלון.
* אפשרות של התראה למטה במידה ונלחץ כפתור על המשך לשאלה הבאה כאשר לא נבחרה תשובה מאחת האפשרויות.

**תפעול המערכת :**

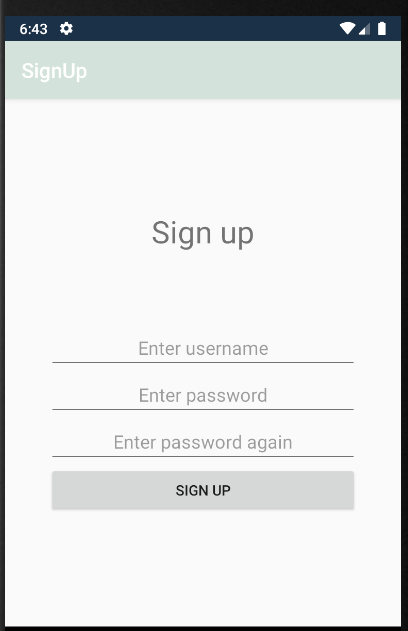
**מסך הפתיחה :**

מסך הפתיחה הוא מסך התחברות לאפליקציה למשתמשים שכבר רשומים אליה. הוא כולל שני שדות להכנסת נתונים : שם משתמש וסיסמה. וכפתור התחברות שמעביר את המשתמש למסך הקטגוריות. במידה ומשתמש לא רשום לאפליקציה, הוא צריך להירשם לפני כניסתו למשחק עצמו לכן יש באפשרותו ללחוץ על כפתור שמעביר אותו למסך ההירשמות וברגע שהוא נרשם, הוא מועבר למסך הקטגוריות.

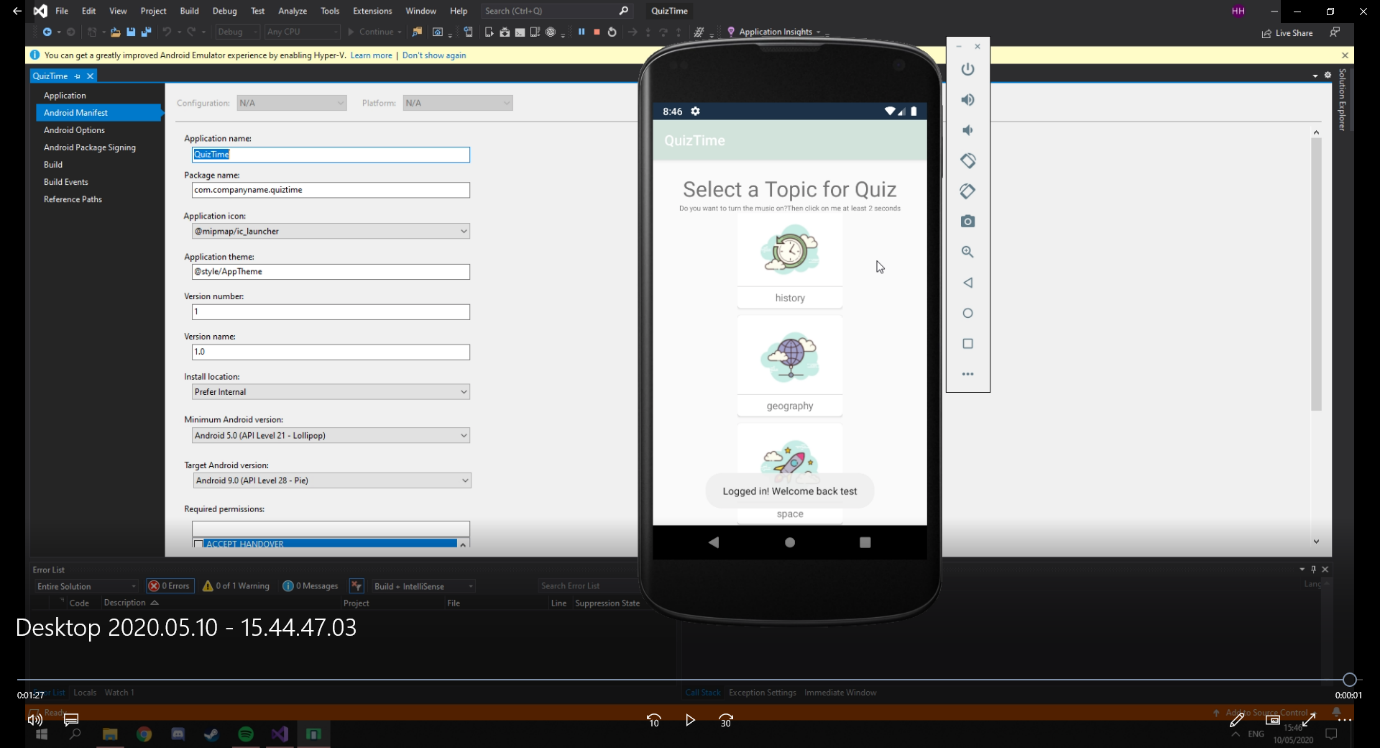


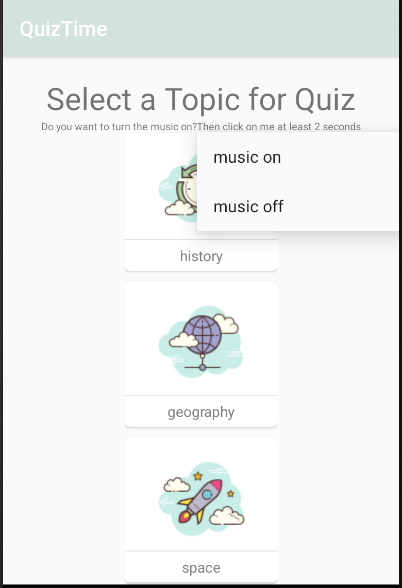
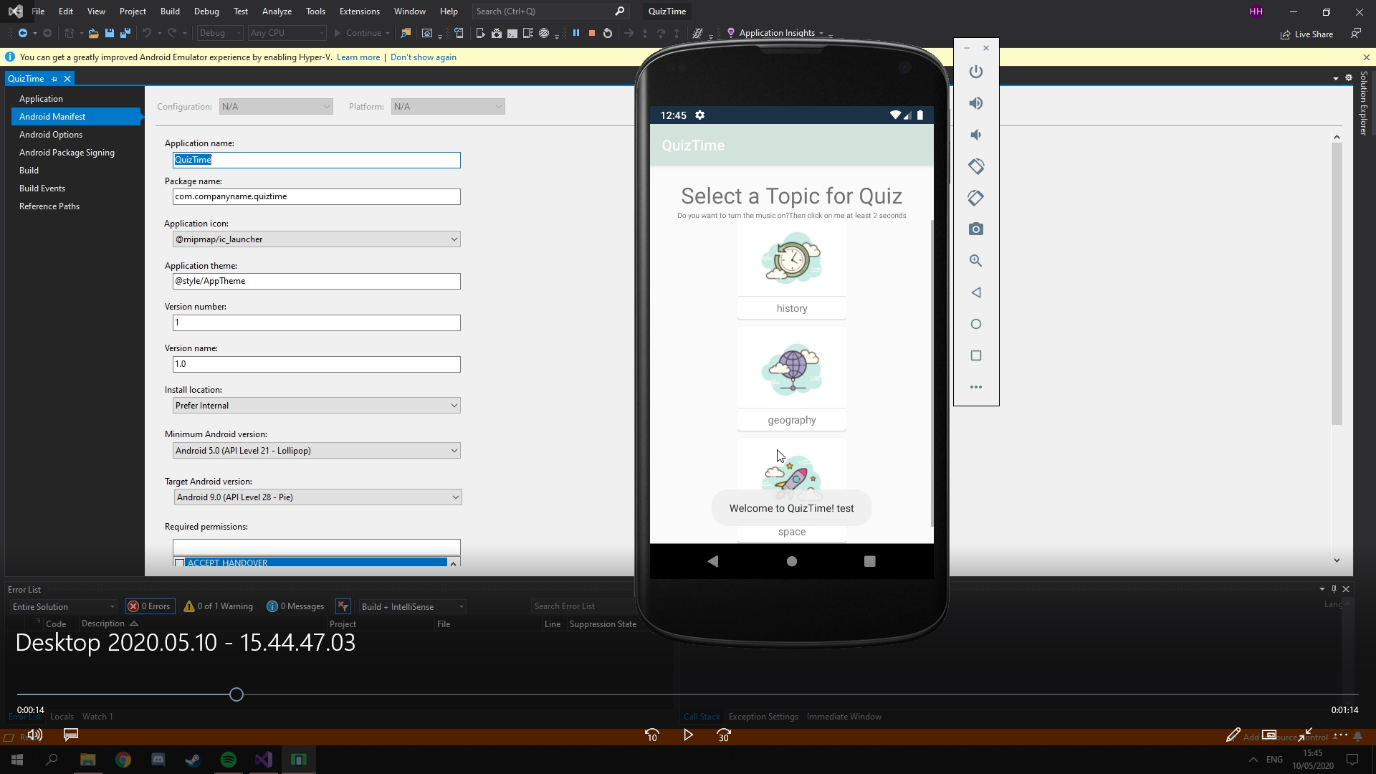
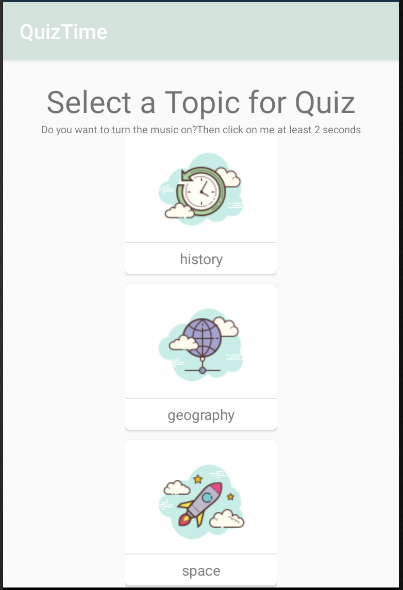
**מסך ההרשמה :**

מסך ההרשמה כולל שלוש שדות להשלמת נתונים בכדי להירשם לאפליקציה, השדות הם שם משתמש, סיסמה ואישור סיסמה. בעת לחיצתו של המשתמש על כפתור השליחה, המשתמש נרשם לאפליקציה ומועבר הישר למסך הקטגוריות.



**מסך הקטגוריות :**

ברגע שמשתמש מתחבר או נרשם לאפליקציה הוא מועבר למסך זה כאשר בחלקו התחתון של המסך מופיעה הודעה האומרת "Welcome to QuizTime!" ושם המשתמש ליד במקרה שנרשם ו"Logged In!Welcome back" ושם משתמש במקרה שהתחבר עם משתמש קיים. החלק העליון הוא כותרת שמסבירה למשתמש שעליו לבחור נושא ומתחת באותיות קטנות ישנה אפשרות של תפריט נשלף בו המשתמש יכול לבחור האם להדליק מוזיקת רקע או לכבות אותה. בשביל שהתפריט יפתח על המשתמש ללחץ 2 שניות על הטקסט. מסך הקטגוריות עצמו הוא בעצם מסך שכולל את נושאי השאלון מוצגים בתור כרטיסיות, כאשר כל כרטיסיה כוללת את התמונה של הקטגוריה ואת שם הקטגוריה למטה. ישנם שלוש קטגוריות סך הכל : גיאוגרפיה, היסטוריה וחלל. מסך זה בעצם מהווה עבור המשתמש יכולת בחירה באיזה נושא ירצה להיבחן.   
כמו כן המסך הוא מסוג scroll view כלומר המשתמש יכול לדפדף למטה. אפשרות זאת נועדה למקרה ויתפסו עוד קטגוריות בעתיד.



הודעת טקסט בעת כניסה לאפליקציה

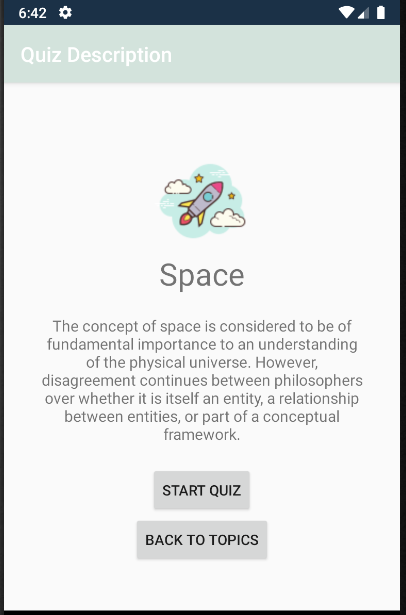
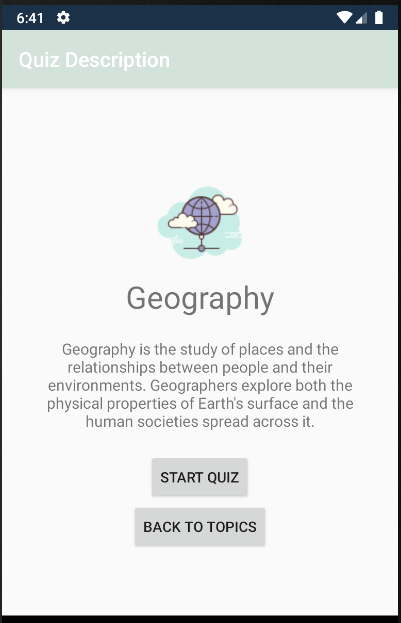
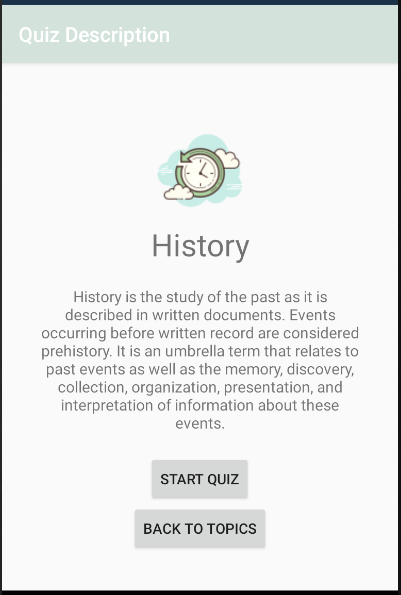
שם משתמשtest\*

תפריט נשלף

בלי אינטראקציה

הודעת טקסט בעת הרשמה לאפליקציה  
ואפשרות דפדוף במסך מצד ימין  
 שם משתמשtest\*

**מסך ההסבר :**

בעת לחיצתו של המשתמש באחת הקטגוריות הוא מועבר למסך ההסבר. מסך זה כולל :  
כותרת שהיא בעצם הנושא שאותו בחר, הסבר קצר על הנושא ושני כפתורים, כפתור המשך לשאלון בנושא שנבחר וכפתור חזרה למסך הקטגוריות במקרה והתחרט.

הסבר על נושא החלל

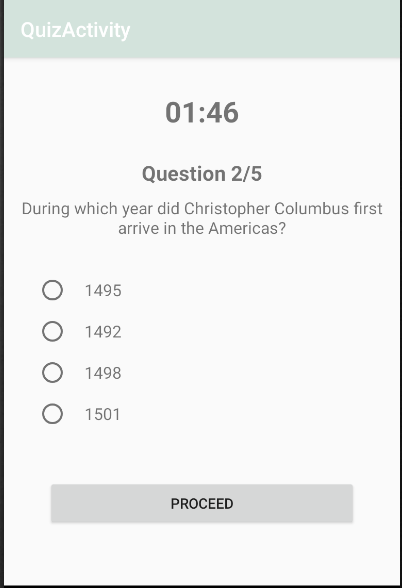
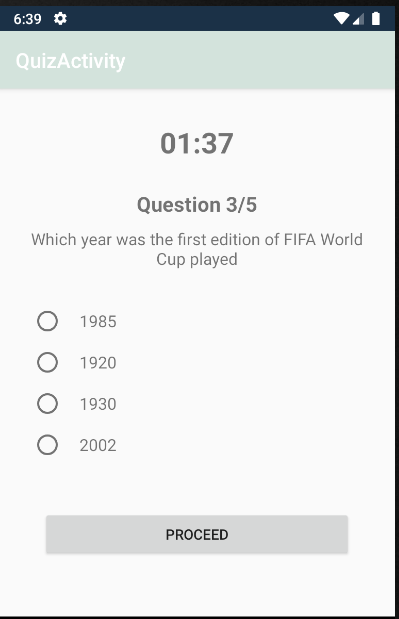
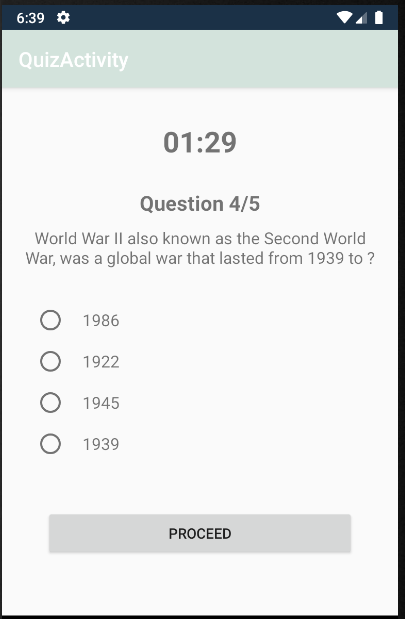
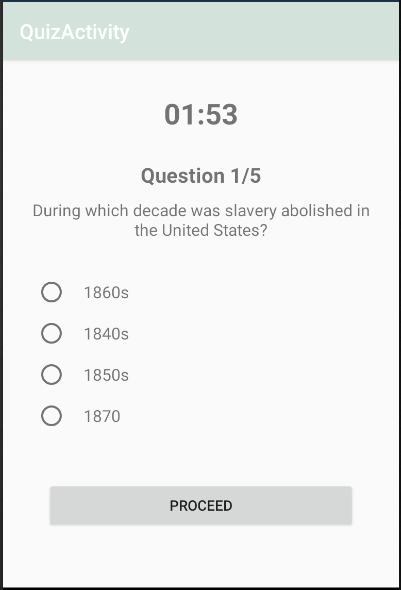
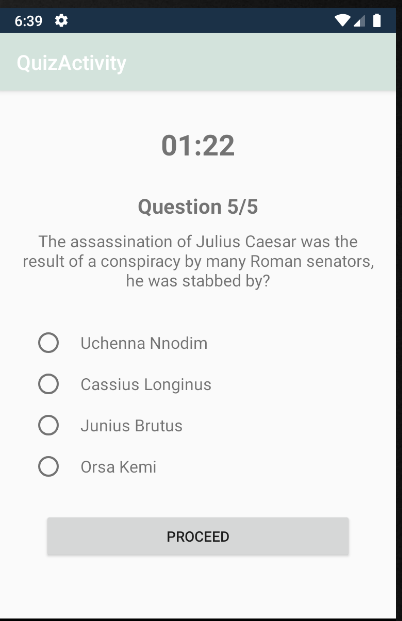
הסבר על נושא הגיאוגרפיה

הסבר על נושא ההיסטוריה

**מסך השאלון :**

מסך השאלון כולל בתוכו טיימר בחלקו העליון של המסך. הטיימר מתחיל משתי דקות וזה זמן המיועד להשלמת כל השאלון בשלמותו. תחת הטיימר מופיע מספור השאלה הנוכחית מתוך סך כל השאלות ומתחת למספור מופיעה השאלה עצמה. יש סך הכל חמש שאלות הנבדלות זו מזו בכל שאלון בנושא המתאים. בהמשך למסך מופיעות ארבע אפשרויות שונות של תשובה בצורת עיגול שניתן ללחיצה (radio button) וכפתור של שליחת השאלה.

**שאלון ראשון – נושא היסטוריה :**

********

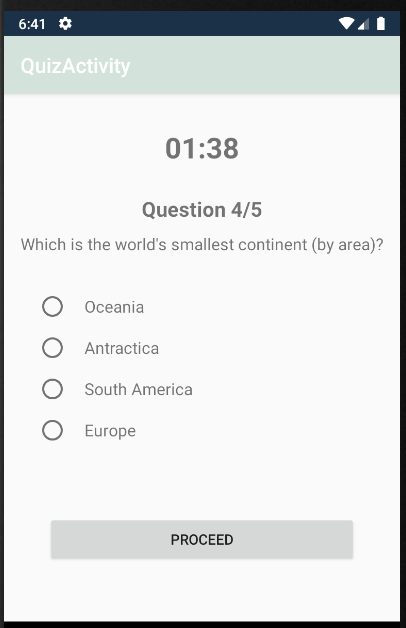
שאלה חמישית

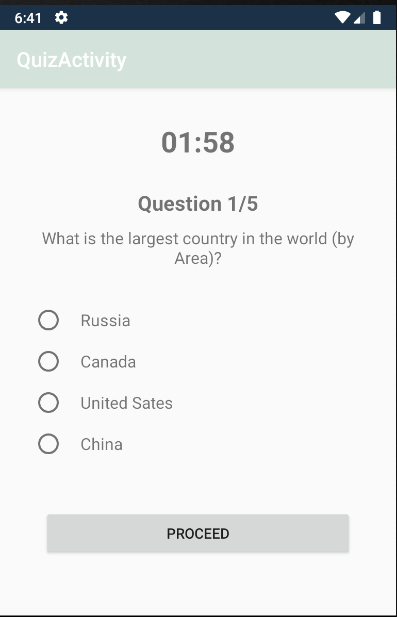
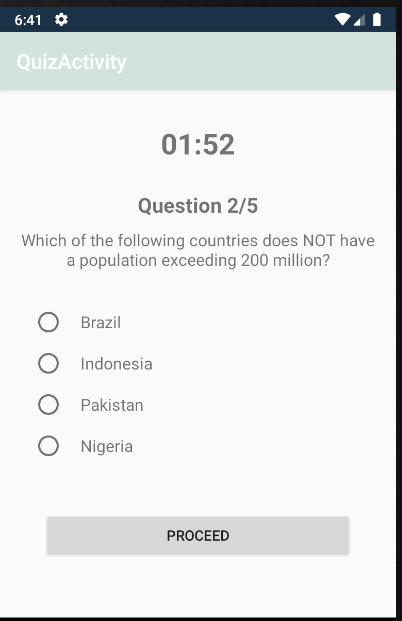
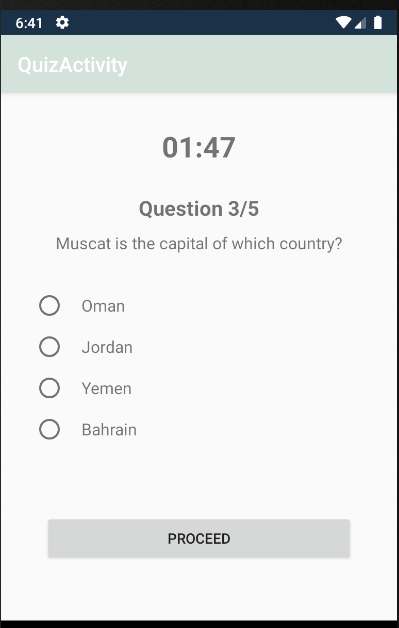
שאלה שנייה

שאלה ראשונה

שאלה רביעית

שאלה שלישית

**שאלון שני – נושא הגיאוגרפיה :**

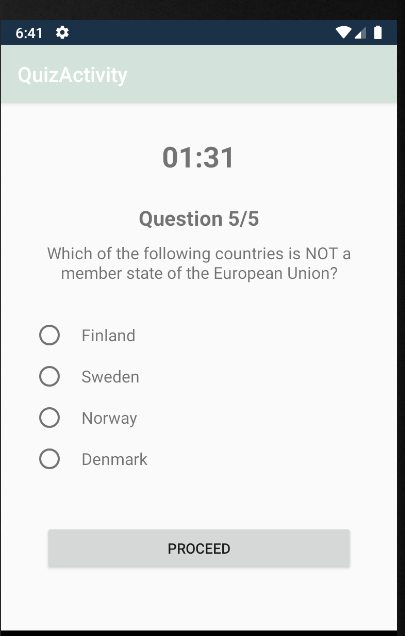


שאלה רביעית

שאלה שלישית

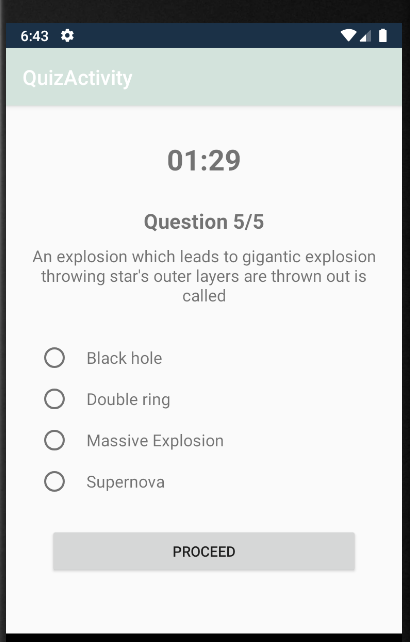
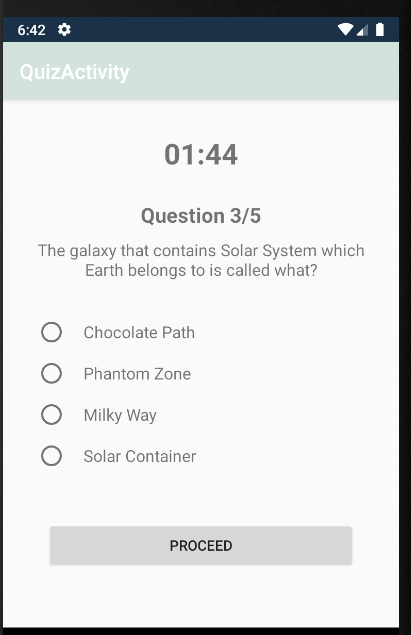
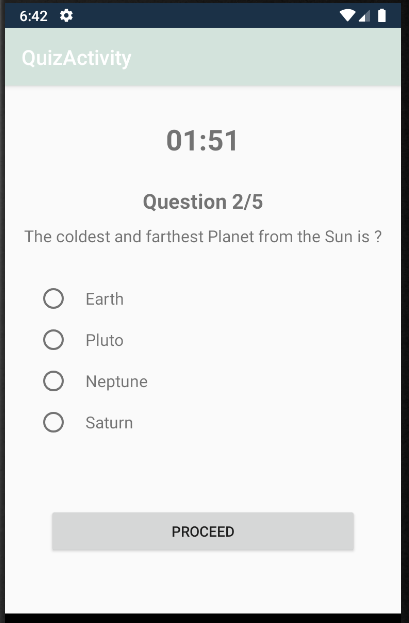
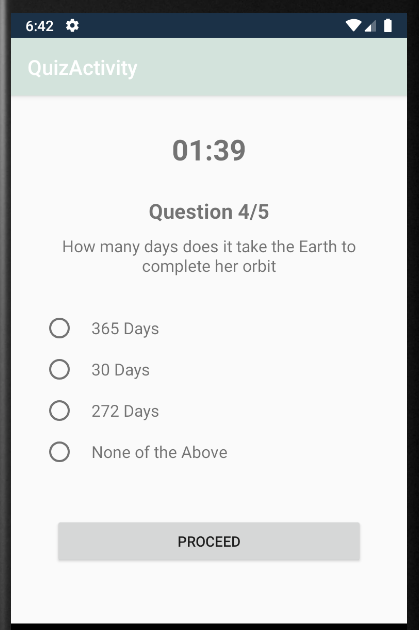
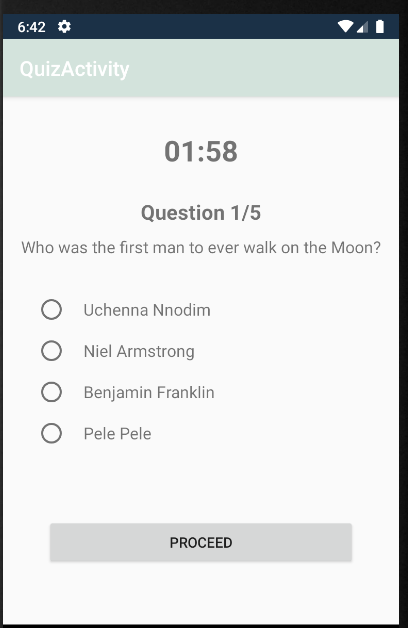
שאלה שנייה

שאלה ראשונה



שאלה חמישית

**שאלון שלישי – נושא החלל :**



שאלה חמישית

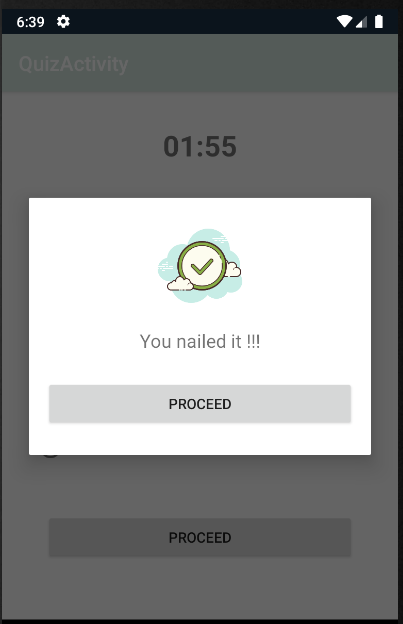
שאלה ראשונה

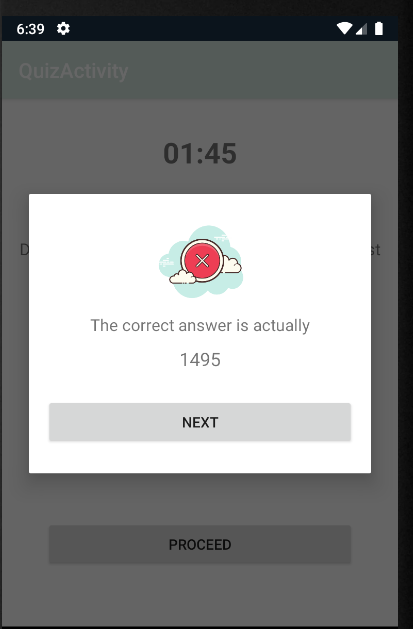
שאלה שלישית

שאלה שנייה

שאלה רביעית

**מסך/חלון קפיצה מסוג Dialog – תשובה נכונה :**

ברגע שהמשתמש עונה תשובה ותשובה זו מתבררת כנכונה, מופיע לפניו חלון עם משפט עידוד של "עשית את זה" עם תמונה של וי וכפתור של מעבר לשאלה הבאה בתור.  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
**מסך/חלון קפיצה מסוג Dialog – תשובה שגויה :**  
  
ברגע שהמשתמש עונה תשובה ותשובה זו מתבררת כשגויה, מופיע לפניו חלון עם התשובה הנכונה לשאלה וכפתור של המשך לשאלה הבאה בתור.

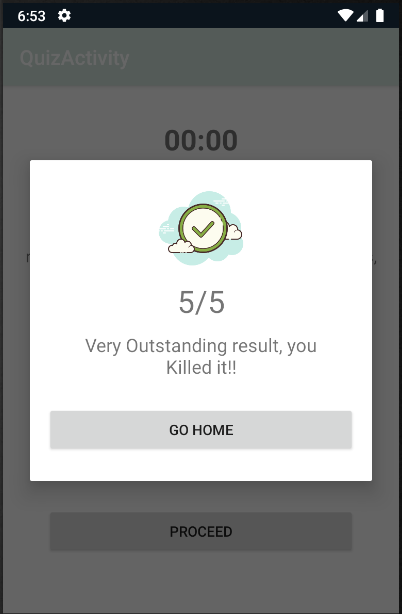
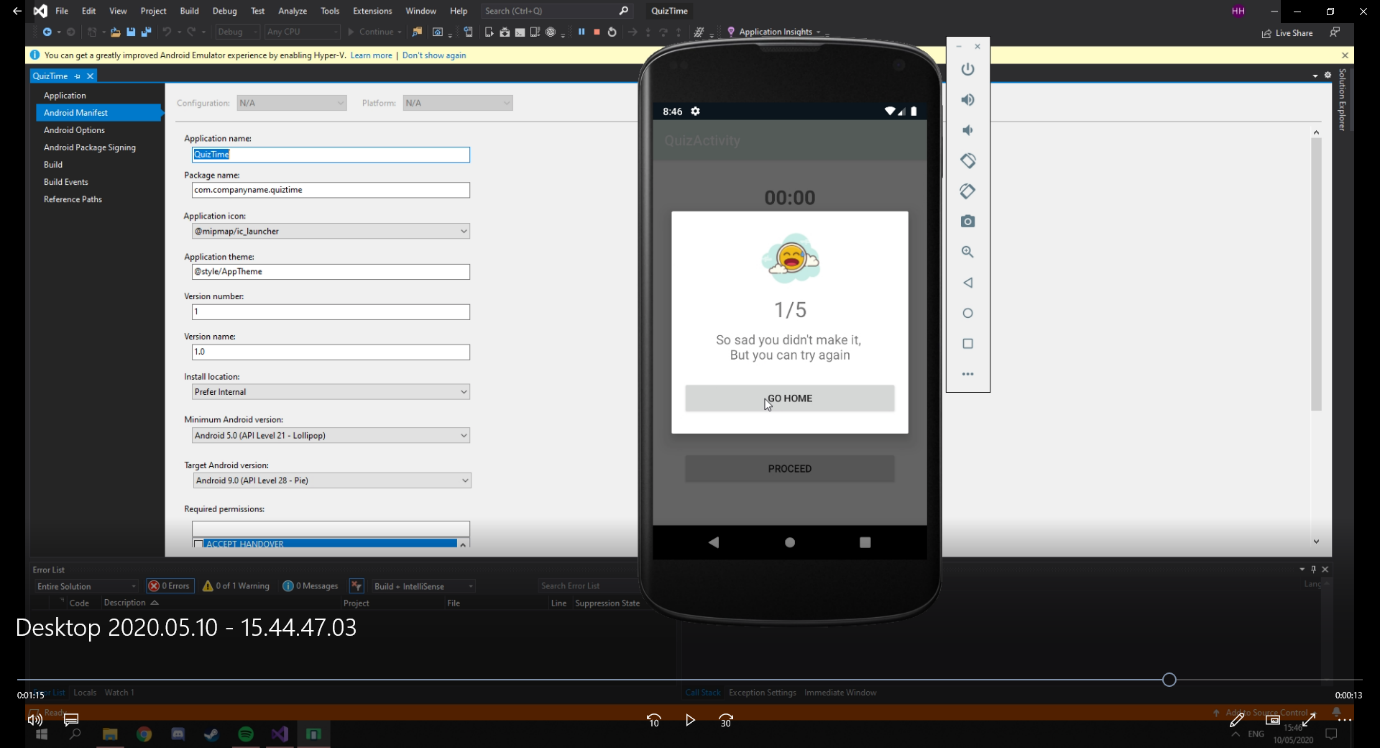
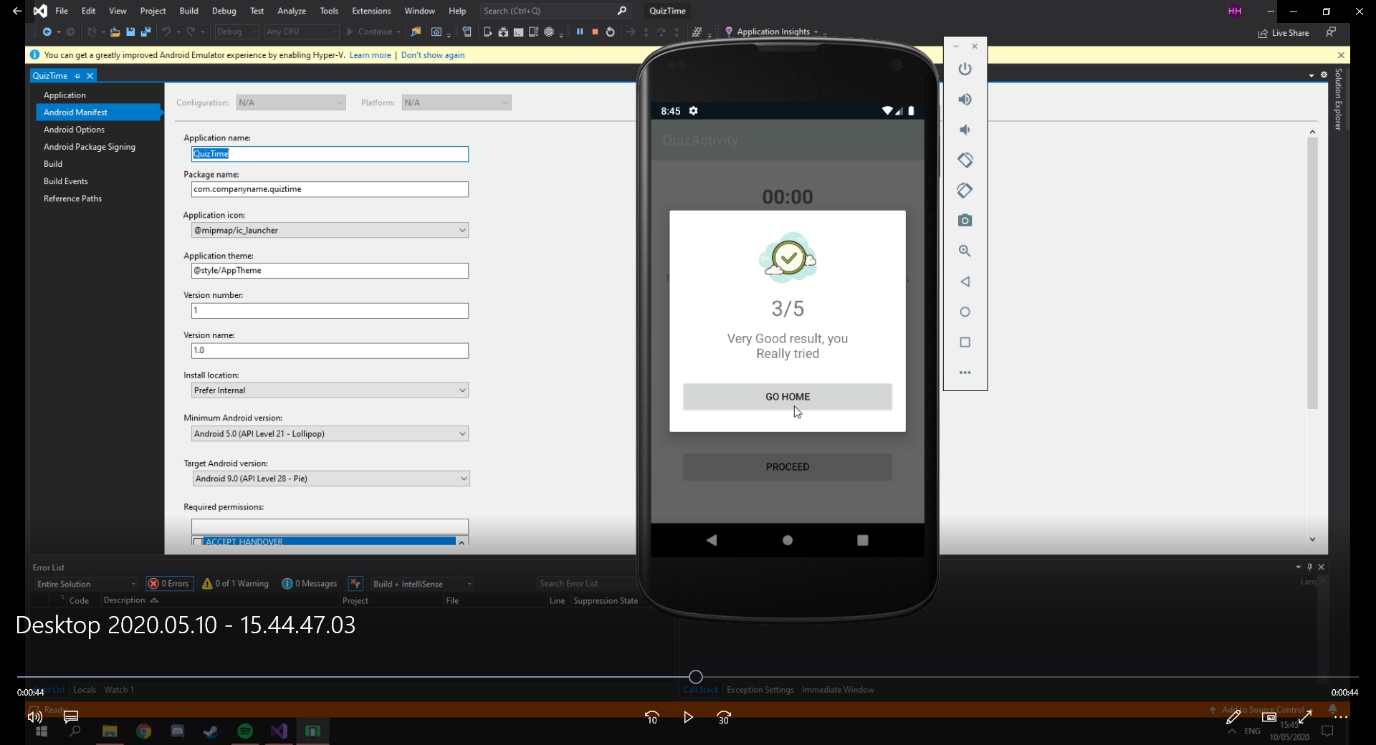


**מסך/חלון קופץ מסוג Dialog של שאלון שהושלם :**

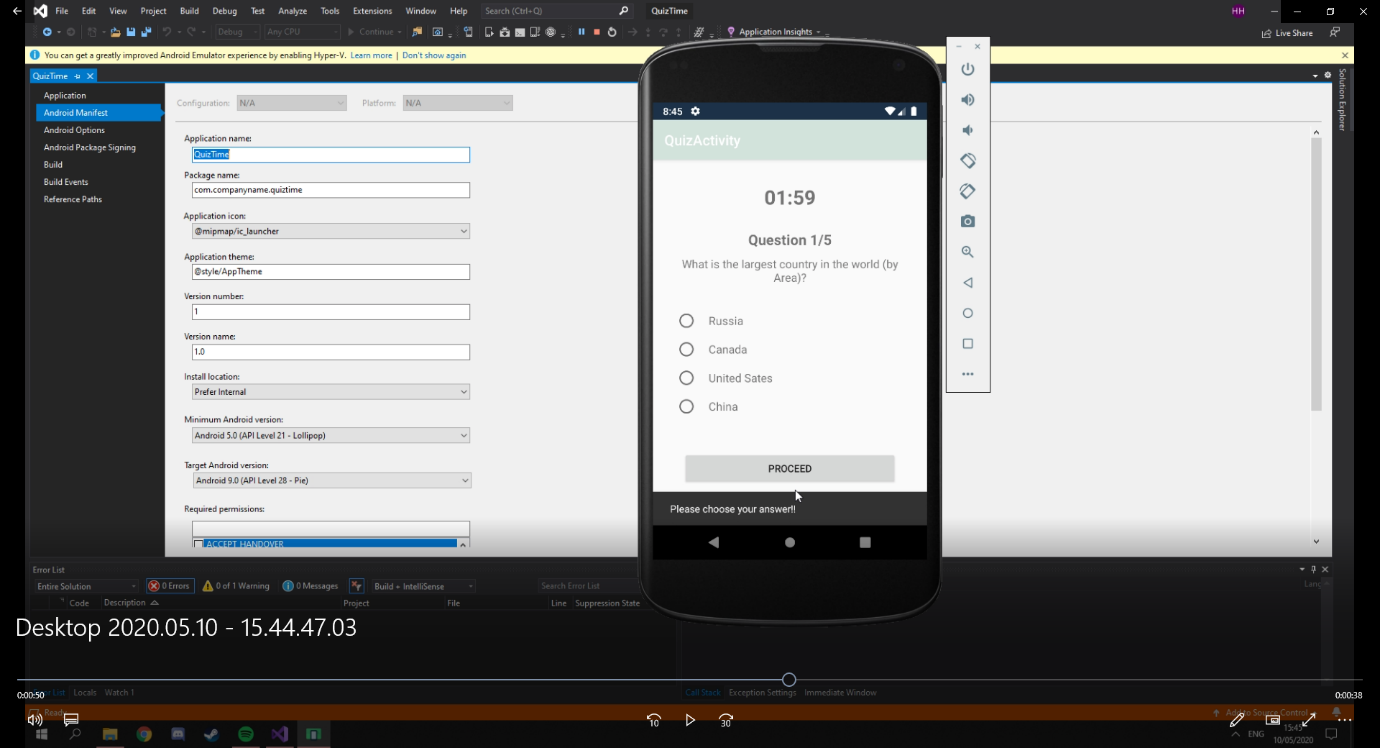
כאשר המשתמש מסיים את השאלון מופיע בפניו מסך סיום עם מספר השאלות הנכונות שהוא ענה מתוך סך השאלות ומשפט סיום התואם לאחוז ההצלחה שלו בשאלון. לדוגמה אם עניתי על שלוש שאלות נכונות מתוך חמש שאלות המשפט יהיה "very good job, you really tried" כלומר עבודה יפה, רואים שהשתדלת מאוד. כמו כן, על המסך יהיה כפתור של חזרה לבית שזו בעצם חזרה לעמוד הקטגוריות.

שאלון שלא ענו בו על אף שאלה, כלומר השאלון התחיל, הזמן רץ אך המשתמש לא נוגע באפשרויות של התשובות יחשב כשאלון שבו ענו על כל השאלות אך כל התשובות היו שגויות כלומר אפס מתוך חמש.

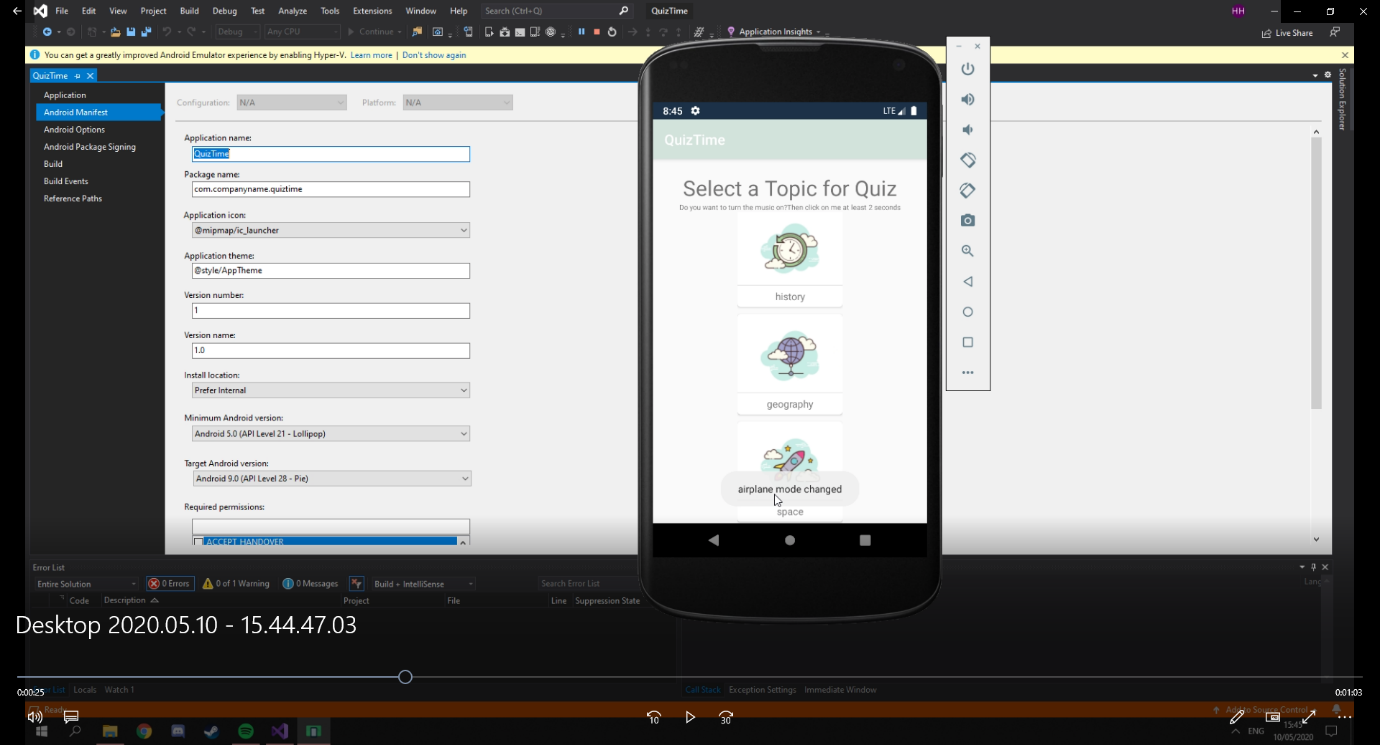
בנוסך למסכים הבאים שהוצגו ישנו משפט שאומר שהתוצאה של השאלון היא ממוצעת אך המספר של השאלות בשאלון שלי הוא אי זוגי ולכן זה בלתי אפשרי שהתשובה תהיה "באמצע".

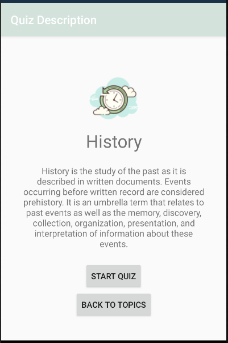
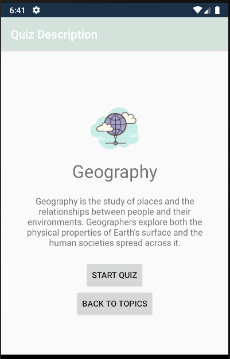
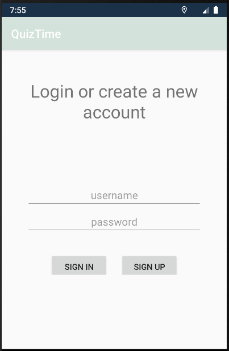
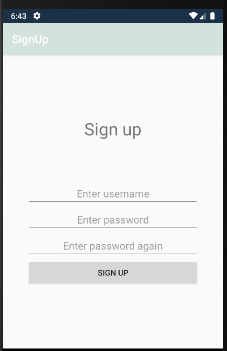


\***הודעת אזהרה :**

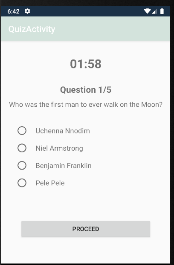
המשתמש לא יכול לעבור לשאלה הבאה בתור בלי שהוא עונה על השאלה שהוא נמצא בה. ברגע שהמשתמש מנסה לעשות זאת מופיעה אזהרה בחלקו התחתון של המסך המבקשת לענות על השאלה.

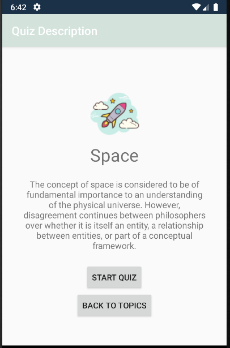
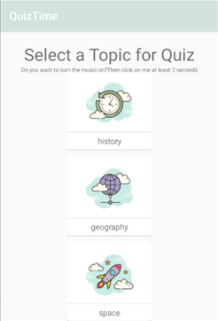
**התראה על שינוי מצב טיסה :**

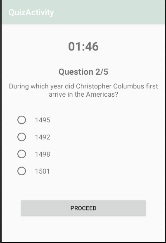
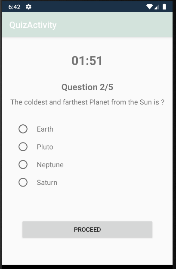
ברגע שהמשתמש מדליק או מכבה את מצב הטיסה, מופיעה באפליקציה התראה המודיעה על כך בתחתית המסך.

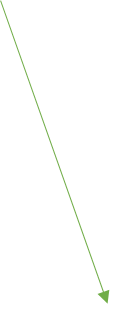
**תרשים מסכים בעבור משתמש נרשם/רשום :**

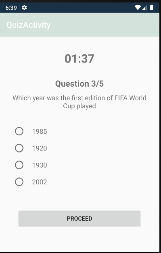
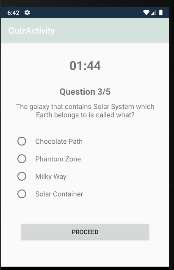
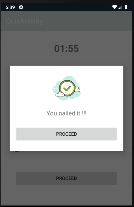
****

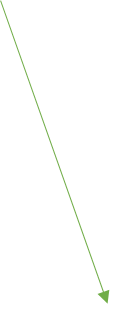


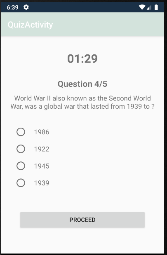
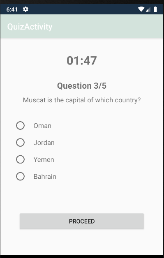
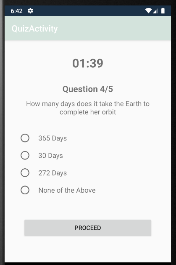
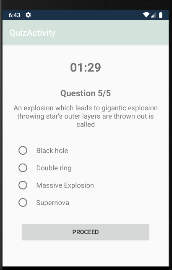
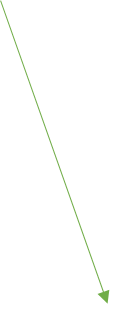
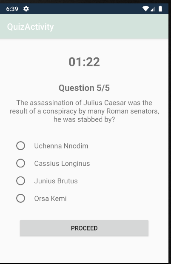
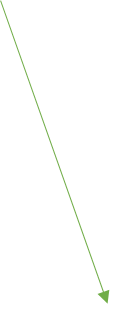
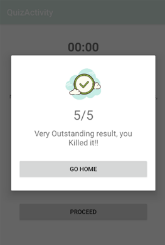
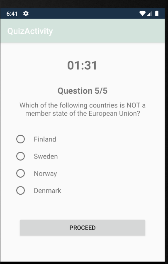
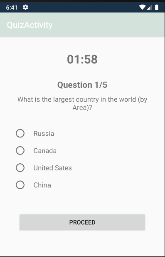
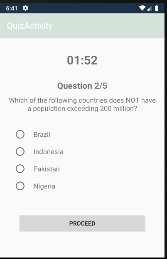
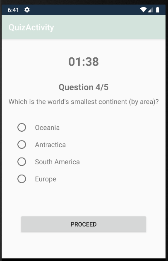












**הרשאות ודרישות מיוחדות :**

* בכדי להתחיל את המשחק על המשתמש קודם להתחבר לאפליקציה או להירשם. בלי זה הוא לא יוכל לעבור למסך הקטגוריות ולהתחיל את השאלון.
* האפליקציה עובדת על כלל המכשירים בעלי מערכת הפעלה של Android.
* [BroadcastReceiver(Enabled = true)] – הרשאה הנמצאת במחלקה FlightModeReceiver המשמשת את הברואדקאסט, במסגרת הרשאה זו מאפשרת האפליקציה שימוש בברואדקאסט.

**גרסת Android מינימלית :**

Android 5.0 API 21

**אילוצים והגבלות :**

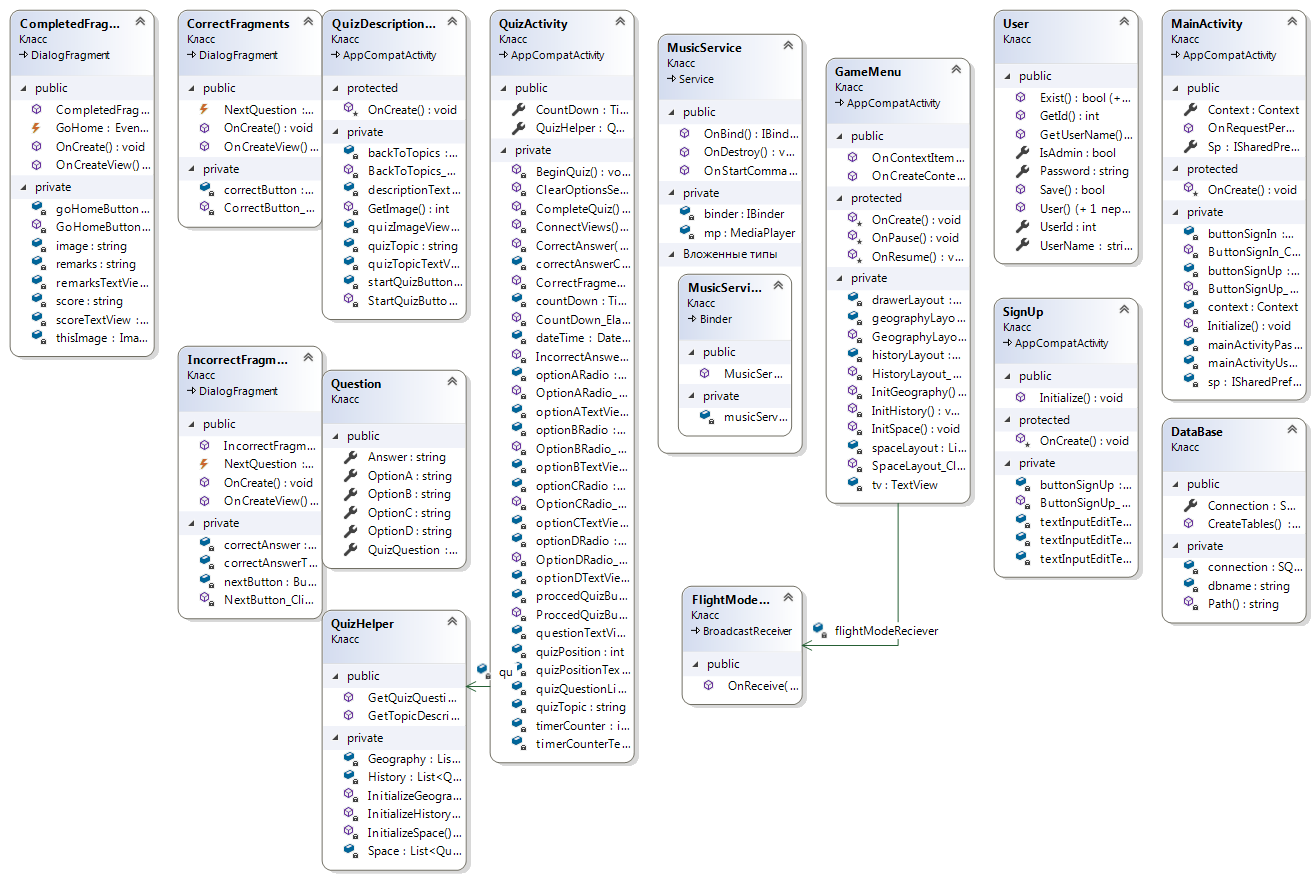
* אילוצי תוכנה : פיתוח בתוכנת מחשב Visual Studio Community 2019 , שימוש במסד נתונים SQLite.
* אילוצי ציוד : מחשב, עכבר, מקלדת.
* אילוצי כתיבה : הכתיבה היא בלשון זכר אך מתייחסת לשני המינים כאחד.

**מדריך למתכנת**

**אתגרים מרכזיים :**

* אתגר מבנה האפליקציה - בעת בניית האפליקציה תכננתי לעשות את השאלות בצורת שלבים אך הבנתי שהרמה של הקוד תהיה גבוהה מידי ומעבר ליכולתי, במיוחד בהתחשב העובדה כי פרץ נגיף קטלני בשם הקורונה שמנע מאתנו – התלמידים – אפשרות לתקשר עם המורה בבית הספר.  
  מבנה כזה של אפליקציה היה יוצר המון אתגרים כמו, הסתרת כל השלבים ושימור השלבים שהמשתמש עבר במסד נתונים. לא יכולתי למצוא גם שום דבר דומה באינטרנט שמסביר כיצד רעיון השלבים עובד מוחשית בקוד בכדי שאוכל לקבל השראה כלשהי שאוכל לממש ולהלביש בצורה כלשהי על הקוד שלי.   
  היות ומבנה כזה של אפליקציה יצר עבורי יותר מחשבות על "איך" ולא על עשייה ורעיונות למימוש, החלטתי לעשות קטגוריות שגם הן היו אתגר.
* אתגר הקטגוריות והשאלון - הקטגוריות דומות בפעילות שלהן לפעילות של השלבים אך הן קבועות ואינן דורשות דינמיקה, כלומר לא תלויות במשתמש (האם הוא עשה את הקטגוריה, השלים אותה עד הסוף וכו' בדומה לשלבים). באינטרנט קיימים המון הסברים למשחקי טריוויה, משום שסוג זה של משחקים הוא מאוד פופולרי וקיבלתי מהסברים אלו עוד המון רעיונות שרציתי להוסיף לאפליקציה שלי כמו רמות קושי לאפליקציה, הוספת ניקוד לשאלות, הוספת כמות גבוהה יותר של שאלות וגם של הקטגוריות וכדומה, אך שוב, בעקבות הקורונה לא הספקתי לממש את כל רעיונותיי. בכל אופן התגברתי על המכשול של מבנה אפליקציה של קטגוריות על ידי כך שכל קטגוריה הייתה ככרטיסיה והעבירה הלאה למסכים הבאים בעזרת מחלקות ששומרות את שם הקטגוריה ובכך מתאימות לה את התיאור של השאלון לפי הנושא,השאלות והתשובות (QuizHelper) ובהמשך לשאלון עצמו כאשר השאלה היא מחלקה Question ששומרת בתוכה את השאלה, אפשרויות תשובה ואת התשובה הנכונה (כמשתנים לא כנתונים עצמם).   
  כלומר הפתרון היה חלוקת החלקים בשאלון לכמה חלקים נפרדים ובסוף לחבר בניהם באקטיביטי הסופי של השאלון עצמו.
* אתגר מסד הנתונים - אתגר נוסף עבורי היה מסד הנתונים SQLite משום שלפני כן לא השתמשתי בו ולא ידעתי כיצד לכתוב את הקוד, לשלוף נתונים או ליצור טבלאות. כמו כן האפליקציה שלי דיי הייתה תלויה במסד הנתונים משום שתחילת המשחק כרוך בהרשמה/התחברות לאפליקציה. התגברתי על מכשול זה על ידי צפייה בסרטוני הסברים ובהסברים של המורה על שליפת הנתונים מהטבלה וכו'.
* אתגר MockView – ברגע שהייתי מנסה לצפות במסכים בקונסטרקטור, במיוחד במסך של הקטגוריות עם הוידג'טים של הכרטיסיות זה לא היה מציג לי אותם וחשבתי שיש לי שגיאה או בעיה כלשהי בקוד למרות שהשתדלתי מאוד לכתוב הכל ביסודיות ובמדויק בכדי שלא יהיו שום שגיאות, וזה העיצוב לכן איזו שגיאה כבר יכולה להיות לי? אתגר זה מאוד התיש אותי ולא ידעתי מה לא בסדר עד שניסיתי להריץ את זה על מחשב של חבר וגיליתי שאצלו הכל פועל כראוי ללא שום MockView . הסתבר שזה בגלל שהמעבד שלי במחשב הוא מסוג amd כאשר ההצגה המובנית בתוך הvisual studio , בין אם זה הקונסטרקטור או האימולטור, עובדים רק על מעבד שהוא intel .
* אתגר הגשת השאלה – במהלך העבודה התלבטתי האם לעשות ניקוד על כל שאלה או לספור את כמות השאלות הנכונות מתוך סך השאלות והאם להודיע להשתמש שהתשובה שהוא ענה לשאלה היא נכונה או שגויה. בסופו של דבר החלטתי ללכת על להודיע ישירות אחרי שהמשתמש מגיש את התשובה האם היא נכונה או לא, מטעמי נוחות וכי גם אני בעצמי אוהבת לדעת ישר מה התשובה. כמו כן, בסופו של דבר האפליקציה באה להרחיב את הידע הכללי של המשתמשים כך שההתרכזות בו היא על נכונות המידע והתשובות לשאלות ולא על ניקוד ותחרותיות, למרות שגם את זה חשבתי להוסיף בסופו של דבר אך מטעמי המצב המדיני והעולמי בעקבות הקורונה לא הספקתי להוסיף זאת.  
  בכל מקרה הדרך שבחרתי בה יצרה עבורי מכשול שהתגברתי עליו בעזרת יצירת שני מסכים correct.xml ו wrong.xml ושני מחלקות שיורשות ממחלקת הדיאלוג (CorrectFragments.cs ו IncorrectFragments.cs )שיוצרות את ההודעה עצמה במסך ובמחלקה QuizActivity.cs בהתאם למצב התשובה (שגויה או נכונה) זה מציג את המסך דיאלוג.

**תוכן :**

**תרשים uml :**

**Activities :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **עיצוב המסך** | **פעולות** | **תכונות** | **תיאור** | **שם בתוכנה** |
|  | protected override void OnCreate(Bundle savedInstanceState) private void Initialize()  private void ButtonSignUp\_Click1(object sender, EventArgs e)  private void ButtonSignIn\_Click(object sender, EventArgs e)  private void ButtonSignIn\_Click(object sender, EventArgs e) | EditText mainActivityUsername, mainActivityPassword;  Button buttonSignIn, buttonSignUp; | מסך הפתיחה | MainActivity |
|  | protected override void OnCreate(Bundle savedInstanceState)  protected override void OnResume()  protected override void OnPause()  public override void OnCreateContextMenu(IContextMenu menu, View v, IContextMenuContextMenuInfo menuInfo)  public override bool OnContextItemSelected(IMenuItem item)  void SpaceLayout\_Click(object sender, EventArgs e)  void GeographyLayout\_Click(object sender, EventArgs e)  void HistoryLayout\_Click(object sender, EventArgs e)  void InitSpace()  void InitGeography()  void InitHistory() | Android.Support.V4.Widget.DrawerLayout drawerLayout;  LinearLayout historyLayout;  LinearLayout spaceLayout;  LinearLayout geographyLayout;  TextView tv; FlightModeReciever flightModeReciever; | מסך הקטגוריות | GameMenu |
|  | private void CountDown\_Elapsed(object sender, ElapsedEventArgs e)  void BeginQuiz()  private void ConnectViews()  private void OptionDRadio\_Click(object sender, EventArgs e)  private void OptionCRadio\_Click(object sender, EventArgs e)  private void OptionBRadio\_Click(object sender, EventArgs e)  private void OptionARadio\_Click(object sender, EventArgs e)  private void ProccedQuizButton\_Click(object sender, EventArgs e)  private void CorrectAnswer()  private void IncorrectAnswer()  private void CorrectFragments\_NextQuestion(object sender, EventArgs e)  private void ClearOptionsSelected()  private void CompleteQuiz() | RadioButton optionARadio, optionBRadio, optionCRadio, optionDRadio;  TextView optionATextView, optionBTextView, optionCTextView, optionDTextView, questionTextView, quizPositionTextView, timerCounterTextView;  Button proccedQuizButton;  List<Question> quizQuestionList = new List<Question>();  readonly QuizHelper quizHelper = new QuizHelper();  string quizTopic;  int quizPosition;  double correctAnswerCount = 0;  int timerCounter = 0;  DateTime dateTime;  System.Timers.Timer countDown = new System.Timers.Timer();  public Timer CountDown { get => countDown; set => countDown = value; }  public QuizHelper QuizHelper => quizHelper; | מסך השאלון | QuizActivity |
|  | protected override void OnCreate(Bundle savedInstanceState)  private void BackToTopics\_Click(object sender, EventArgs e)  int GetImage(string topic)  private void StartQuizButton\_Click(object sender, EventArgs e) | TextView quizTopicTextView;  TextView descriptionTextView; ImageView quizImageView;  Button startQuizButton; Button backToTopics;  string quizTopic; | מסך ההסבר על הקטגוריה שנבחרה | QuizDescriptionActivity |
|  | protected override void OnCreate(Bundle savedInstanceState)  public void Initialize()  private void ButtonSignUp\_Click(object sender, EventArgs e) | EditText textInputEditTextName, textInputEditTextPassword, textInputEditTextPasswordAgain;  Button buttonSignUp; | מסך ההרשמה | SignUp |

**Services :**

|  |  |  |  |
| --- | --- | --- | --- |
| פעולות | תכונות | תיאור | שם הסרוויס |
| public override StartCommandResult OnStartCommand(Intent intent, StartCommandFlags flags, int startId)  public override void OnDestroy()  public override IBinder OnBind(Intent intent) | IBinder binder; MediaPlayer mp; | מחלקה זו שמהווה סרוויס היא מחלקה שבעזרתה המשתמש יכול להפעיל מוזיקה שתפעל לאורך כל הזמן מאז שהוא לחץ על כפתור ההדלקה, כמו כן אם ירצה הוא יוכל לכבות את המוזיקה בעזרת כפתור כיבוי. | MusicService |

**Threads :**

בפרויקט שלי לא השתמשתי בטרדס.

**Broadcast Receiver :**

|  |  |  |  |
| --- | --- | --- | --- |
| פעולות | תכונות | תיאור | **שם הברואדקאסט רסיוור** |
| public override void OnReceive(Context context, Intent intent) |  | מחלקה זו שמהווה ברואדקאסט רסיוור מגדירה פעולה שמודיעה למשתמש בעזרת הודעת טוסט על כך שמצב הטיסה השתנה, כלומר או נכבה או נדלק. | FlightModeReciever |

**טבלת מחלקות :**

|  |  |  |  |
| --- | --- | --- | --- |
| פעולות | תכונות | תיאור | שם המחלקה |
| public override void OnCreate(Bundle savedInstanceState)  public CompletedFragment(string \_remarks, string \_score, string \_image)  public override View OnCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)  private void GoHomeButton\_Click(object sender, EventArgs e) | ImageView thisImage;  TextView remarksTextView;  TextView scoreTextView;  Button goHomeButton;  string remarks, score, image;  public event EventHandler GoHome; | מחלקת השאלון הגמור היא מחלקה היורשת מדיאלוג והיא בעצם מהווה את החלון שמופיע בסוף השאלון, המציג תמונה, משפט וניקוד לפי התוצאות של מי שנבחן וכפתור מעבר בחזרה למסך הקטגוריות. | CompletedFragment |
| public override void OnCreate(Bundle savedInstanceState)  public override View OnCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) private void CorrectButton\_Click(object sender, EventArgs e) | Button correctButton;  public event EventHandler NextQuestion; | מלקת החלק הנכון היא מחלקה שיורשת מדיאלוג ומהווה חלון שקופץ בסוף כל שאלה שתשובתה התבררה כנכונה וכוללת את כפתור המעבר לשאלה הבאה. | CorrectFragment |
| private static string Path()  public static void CreateTables()  public static SQLiteConnection Connection | private static readonly string dbname = "userregDb";  private static readonly SQLiteConnection connection = new SQLiteConnection(Path()); | מחלקת הדטה בייס היא המחלקה שיוצרת את הטבלה של .SQLite  ויוצרת את המסלול והחיבור למסד הנתונים. | DataBase |
| public override void OnCreate(Bundle savedInstanceState)  public IncorrectFragments(string answer)  public override View OnCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)  private void NextButton\_Click(object sender, EventArgs e) | string correctAnswer;  TextView correctAnswerTextView;  Button nextButton;  public event EventHandler NextQuestion; | מחלקת החלק השגוי היא מחלקה היורשת מדיאלוג ומהווה חלון שקופץ בסוף כל שאלה שהייתה שגויה וכוללת את כפתור המעבר לשאלה הבאה. | IncorrectFragment |
| public string QuizQuestion { get; set; }  public string OptionA { get; set; }  public string OptionB { get; set; }  public string OptionC { get; set; }  public string OptionD { get; set; }  public string Answer { get; set; } |  | במחלקת השאלה מוגדרות השאלה, ארבעת אפשרויות לתשובה והתשובה עצמה. | Question |
| public string GetTopicDescription(string topic)  public List<Question> GetQuizQuestions(string topic)  void InitializeHistory()  void InitializeGeography()  void InitializeSpace() | List<Question> History;  List<Question> Geography;  List<Question> Space; | מחלקת העזר לשאלון היא בעצם מחלקה שכוללת בתוכה את כל המידע של השאלון עצמו כלומר את שם הקטגוריה,תיאור הקטגוריה, את רשימת השאלות של כל הקטגוריות ואת התשובות אליהן. | QuizHelper |
| public int UserId { get; set; }  public string UserName { get; set; }  public string Password { get; set; }  public bool IsAdmin { get; set; }  public User()  public User(string userName, string password)  public static bool Exist(string userName)  public static bool Exist(string userName, string password)  public static int GetId(string userName, string password)  public static string GetUserName(int id)  public static bool Save(string userName, string password) |  | מחלקת המשתמש היא מחלקה שיוצרת את המשתמש ומקבלת את הנתונים שלו שהם שם המשתמש והסיסמה וכמו כן שולפת נתונים מהטבלה במסד הנתונים. | User |

**שיקולי תוכן – Design :**

ישנים מספר שיקולי תוכן בהם השתמשתי במסגרת עבודתי על הפרויקט מסיבות כאלו ואחרות והם :

* חלוקה למחלקות – כבר בתחילת העבודה הבנתי שעליי לחלק את הפרויקט למחלקות נפרדות בכדי להקל על העבודה. חלוקה למחלקות עוזרת לשמור על סדר ולא מבלגנת את הכל כך שבאמצע הפרויקט אני אסתכל על כל מה שכתבתי ולא אבין מה זה מה.

כמו כן השימוש במחלקות נורא קל משום שבקלות ניתן לדעת איזו מחלקה שייכת לאיזה חלק בפרויקט ולמה היא משתמשת מה שמקל מאוד וחוסך זמן.   
מעבר לכך, חלוקה למחלקות מונעת כפילויות כך שלא אצטרך לכתוב את אותה פעולה כל פעם מחדש משום שכבר יש לי אותה מוגדרת במחלקה ואני רק צריכה לקרוא לה.

* שמות משתנים – אני חושבת שמסגרת העבודה החלק הכי חשוב בה היה נתינת שמות למשתנים לפי סוג המשתנה ולפי ההשתייכות שלו. דבר זה מאוד הקל עליי מלא להתבלבל בין משתנים זהים בסוגם אך כאשר הפעולה שלהם היא שונה. זה חוסך זמן בעתיד מלחפש מה כל כפתור אומר ואיפה הוא נמצא.
* חלוקה לתיקיות – שזה משהו שלא עשיתי אך לדעתי זה היה מאוד מקל את עבודתי על הפרויקט עצמו וגם על הספר. חלקה של כל הפרויקט לתיקיות שונות, לדוגמה תיקיית מחלקות עזר לשאלון, תיקיה למסד הנתונים, הירשמות והתחברות. זה היה מאוד מסדר את עבודתי כך שלא הייתי צריכה כל פעם לקרוא מההתחלה עד הסוף את כל רשימת המחלקות והאקטיביטיס והייתי יכולה רק לגשת לתיקיה לפי השם.

**Resources :**

|  |  |  |
| --- | --- | --- |
| שם ה Layout | ה Activity אליו שייך | תיאור |
| activity\_main | MainActivity | מסך הפתיחה הראשי הכולל את העמוד התחברות, שני שדות לשם משתמש וסיסמה וכתפור להתחברות המוביל למסך הקטגוריות ועוד כפתור נוסף במידה והמשתמש רוצה להירשם לאפליקציה. כאשר המשתמש מתחבר מופיעה הודעת טוסט בחלקו התחתון של המסך המברכת אותו על חזרתו לאפליקציה. |
| completed | CompletedFragment זהו קלאס הדיאלוג שמציג את הלייאאוט הזה אך בפועל מתבטא בQuizActivity | חלון הדיאלוג המופיע בסוף שאלון שהושלם. |
| correct | CorrectFragments  זהו קלאס הדיאלוג שמציג את הלייאאוט הזה אך בפועל מתבטא בQuizActivity | חלון הדיאלוג הקופץ לאחר הגשת השאלה שהתבררה כנכונה. |
| gameMenu | GameMenu | מסך הקטגוריות בו יש שלוש קטגוריות שהן היסטוריה גיאוגרפיה וחלל. כמו כן עמוד יש תפריט קונטקסט נשלף שבו מוצגת בפני המשתמש אפשרות להדלקת וכיבוי מוזיקה. בעת לחיצתו של המשתמש על אחת מהקטגוריות הנתונות הוא מועבר למסך ההסבר על הקטגוריה שבחר. |
| include\_main\_page | gameMenu | מסך הקטגוריות שבו מעוצב כל הלייאאוט שמוצג בgameMenu.  דף זה נועד בכדי להלביש עליו עוד עיצוב ווידג'טים בעתיד, מטעמי נוחות אך בפועל אין לו כל שימוש. |
| quiz\_page | QuizActivity | עמוד השאלון, עמוד זה הוא בעצם חיבור של כמה פעולות ממחלקות שונות שמפעילות את השאלון המורכב משאלות, תשובות ואפשרויות לתשובה בהתאם לקטגוריה שנבחרה. באמצע השאלון המשתמש יכול לראות בחלקו העליון של המסך את הטיימר המציג את הכמות המקסימלית להשלמת השאלון כולו שיורד ככל שהזמן עובר.כמו כן כאשר המשתמש מגיש את השאלה, אם היא נכונה קופץ חלון דיאלוג שמעיד על כך וכפתור מעבר לשאלה הבאה, ואם השאלה שגויה קופץ חלון דיאלוג המציג את התשובה הנכונה לשאלה ואת כפתור המעבר לשאלה הבאה. בסוף השאלון קופץ חלון דיאלוג שמציג את כמות התשובות הנכונות מתוך סך השאלות, משפט בהתאם לתוצאה (החישוב מתבצע לפי חלקה של כמות התשובות הנכונות בסך השאלות והמרה לאחוזים) וכפתור חזרה לעמוד הקטגוריות. |
| selected\_topic | QuizDescriptionActivity | מסך הקטגוריה שנבחרה, מסך זה בעצם מהווה מסך הסבר קצר על הקטגוריה שנבחרה, כפתור המשך לשאלון עצמו וכפתור חזרה למסך הקטגוריות במקרה שהמשתמש שינה את דעתו. |
| signup | SignUp | דף ההירשמות למשתמשים שאינם קיימים במערכת. שדות לרשימת שם משתמש סיסמה ואימות סיסמה. כפתור המעביר למסך הקטגוריות. בעת כניסתו מופיעה הודעת טוסט בחלק התחתון של המסך המברכת אותו על הצטרפותו לאפליקציה עם השם משתמש שלו. |
| wrong | IncorrectFragments  זהו קלאס הדיאלוג שמציג את הלייאאוט הזה אך בפועל מתבטא בQuizActivity | חלון הדיאלוג הקופץ כאשר התשובה שהוגשה היא תשובה שגויה. |

**Menu :**

|  |  |  |  |
| --- | --- | --- | --- |
| שם התפריט | סוג התפריט | האקטיביטי בו התפריט מומש | תיאור |
| main\_menu | Options menu | gameMenu | התפקיד מופיע במסך הקטגוריות. הוא נמצא מתחת לכותרת הראשית ונפתח לאחר שלוחצים עליו לפחות שתי שניות.  בתפריט ישנם שני אייטמים, הראשון משמש להדלקת המוזיקה והשני משמש לכיבוי המוזיקה. |

**Drawables :**

|  |  |  |
| --- | --- | --- |
| שם התמונה | תיאור | התמונה |
| correct | התמונה משמת כתמונה שמופיעה בחלון הדיאלוג כאשר נשלחת תשובה נכונה ובמסך הסיום כאשר אחוז ההצלחה של המשתמש בשאלון היא מעל חמישים. |  |
| wrong | התמונה משמשת כתמונה שמופיעה בחלון הדיאלוג כאשר נשלחת תשובה שגויה. |  |
| sad | התמונה משמשת כתמונה שמופיע בחלון הדיאלוג בסוף השאלון כאשר אחוז ההצלחה בשאלון הוא מתחת לחמישים אחוז. |  |
| geography | התמונה משמשת כתמונה במסך הקטגוריות בנושא הגיאוגרפיה  וכתמונה במסך ההסבר של קטגוריית הגיאוגרפיה. |  |
| history | התמונה משמשת כתמונה במסך הקטגוריות בנושא ההיסטוריה  וכתמונה במסך ההסבר של קטגוריית ההיסטוריה. |  |
| space | התמונה משמשת כתמונה במסך הקטגוריות בנושא החלל  וכתמונה במסך ההסבר של קטגוריית החלל. |  |

**Raw :**

|  |  |
| --- | --- |
| שם השיר | תיאור |
| Pyre | השם משמש כשיר רקע ברגע שהמשתמש מדליק אותו במסך הקטגוריות בתפריט הנשלף. |

**טבלת קבצים :**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| שם הטבלה | שיטת השמירה | תיאור | שם המחלקה | תכונות | פעולות |
| tblUsers | SQLite | טבלה זו מהווה את מקום השמירה לפרטים של המשתמשים שנרשמים.  במחלקה זו מוגדרות פעולות שנועדו לשליפת נתונים של המשתמש ופרטיו ולשם הוספת משתמש חדש. | User | public int UserId  public string UserName  public string Password  public bool IsAdmin | public User()  public User(string userName, string password)  public static bool Exist(string userName)  public static bool Exist(string userName, string password)  public static int GetId(string userName, string password)  public static string GetUserName(int id)  public static bool Save(string userName, string password) |

**סכמת DataBase :**

TblUsers

|  |
| --- |
| UserId |
| Username |
| Password |

**פרוטוקול תקשורת :**

במסגרת העבודה על הפרויקט נעשה שימוש במספר חבילות נתונים (packages) :

* sqlite-net-pcl – חבילת נתונים לצורך מסד הנתונים, לשמירת נתונים לטווח ארוך בטבלאות.

**סיכום :**

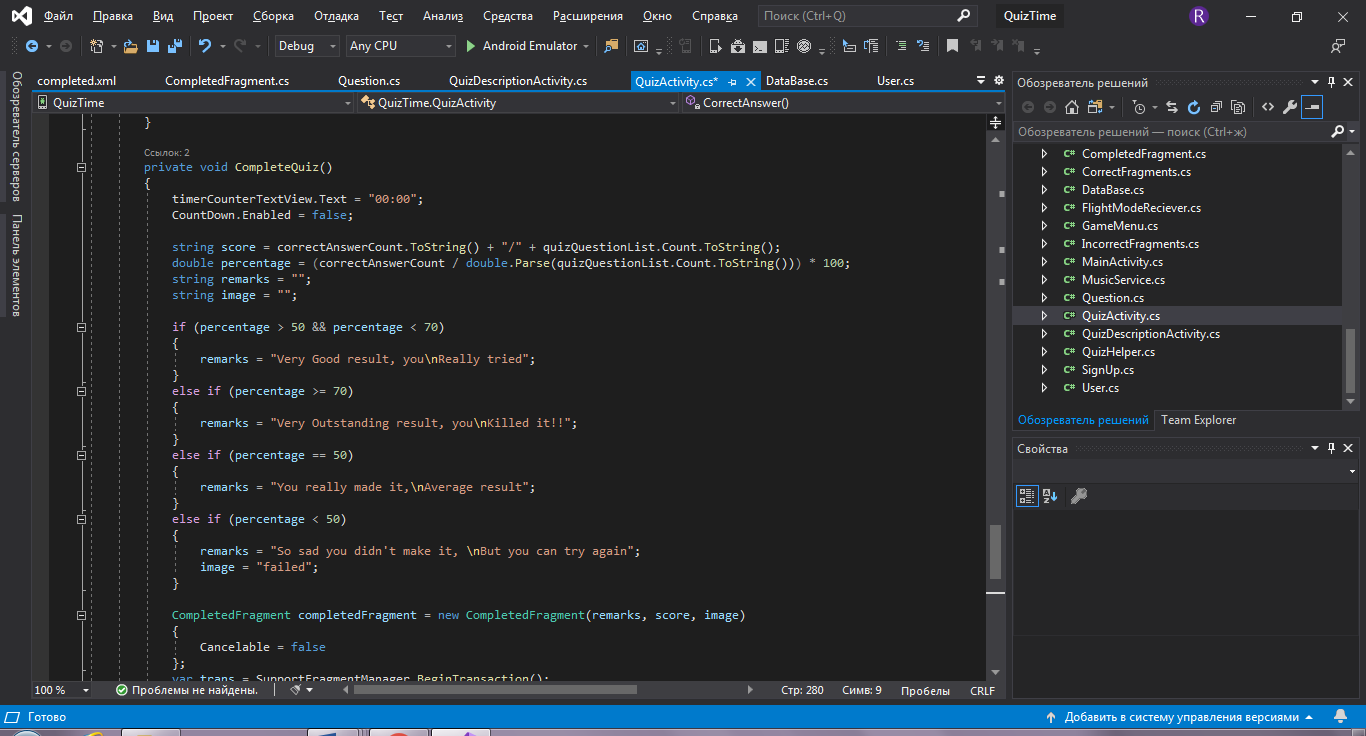
העבודה על הפרויקט הייתה מרתקת ומעניינת, למדתי רבות על השימוש בVisual Studio Xamarin . הודות לפרויקט זה שיפרתי רבות את הידע שלי בתכנות, את יכולת התכנות שלי, יכולת חיפוש, שהתבטאה בעבודה על הפרויקט בחיפוש פתרון לשגיאות שונות שצצו פה ושם לאורך הפרויקט. בנוסף לכך, למדתי לעבוד על כמה חלונות שונים בו זמנים, בפרויקט שלי היו מחלקות רבות שבסופו של דבר הייתי צריכה לקשר בניהן ולהפוך אותן למשהו אחד ושלם. זה גזל ממני המון זמן ומחשבה והסיפוק מהעבודה הגמורה והשלמה שעובדת כראוי היה בלתי ניתן להסבר.

פעמים רבות עלה במחשבתי לוותר, משום שכפי שציינתי בהתחלה, הפרויקט התקלקל לי פעמים רבים והייתי צריכה להתחיל אותו מההתחלה שוב ושוב, דבר שהתיש מאוד. כמו כן, בעקבות האי עבודה הנורמלית של המחשב כי המעבד שלי היה שונה ממה שדרוש לVisual Studio לא התאפשר לי לעבוד בנוחות והייתי זקוקה לפנות לחברים שיריצו את האפליקציה על המחשב שלהם.

באופן כללי היו לי שגיאות רבות שדווקא קשורות לעבודה של התוכנה עצמה ולא לעבודה של האפליקציה שלי ולקח לי זמן להבין זאת. נכון שבסוף התהליך עובדה זו מנחמת אך זה הפך את התהליך לקשה מאוד.

אני מאוד מרוצה מהתהליך שעברתי במהלך העבודה שלי על הפרויקט ועל כך שהצלחתי לסיים את האפליקציה עד הסוף ולממש את רוב התוכניות שלי. מצב הקורונה העולמי דיי הפחיד והלחיץ אותי כי זה שיבש את תהליך העבודה בבית הספר והתוכניות ומנע ממני את האפשרות של העבודה על הפרויקט בבית הספר שבו אמנם המחשבים הם לא ברמה מי ישמע מה, אך לפחות אני יכולה להריץ ולהפעיל עליהם את האפליקציה שלי.

לסיכום, אני לקחתי מהפרויקט הזה המון, בין אם זה לחיים עצמם ובין אם זה לעולם התכנות.   
פרויקט זה היה הפרויקט הראשון שלי בתכנות אפליקציות, אך מה שבטוח – זה שהוא לא האחרון .  
אני מודה לכל האנשים שעזרו לי במהלך הפרויקט, למורה שחר ערוסי, לחברי הכיתה ולחברים מחוץ לכיתה שעזרו לי ותמכו בי ברגעים של התייאשות והרמת ידיים!

**נספחים :**

הצגת הערה בהתאם לאחוז ההצלחה

חישוב אחוז ההצלחה בשאלון

איפוס טיימר בסוף שאלון

שם הפעולה

הפעולה מחשבת את אחוז ההצלחה של המשתמש בשאלון על ידי חילוק של מספר התשובות הנכונות שהוא ענה על ידי מונה שהוגדר לפני כן במספר סך השאלות הקיימות בשאלון, ולפי אחוז מוצגת למשתמש בסוף השאלון הערה ותמונה. במידה ואחוז ההצלחה היה מתחת לחמישים אחוז התמונה תהיה של אימוג'י עצוב, בכל מצב אחר תמונה של וי.

**ביבליוגרפיה :**

* appschool - אתר עם סרטוני הסברים בתשלום בVisual Studio 2019 .

קישור : appschool.co.il

* Stack Overflow – אתר המכיל שאלות על תכנות בשפות שונות ופלטפורמות שונות.  
  קישור : stackoverflow.com
* Microsoft – Xamarin – אתר המכיל הסברים שונים על מחלקות שונות באנדרואיד והשימוש בהם.
* קישור : <https://docs.microsoft.com/en-us/xamarin/>