סדנא

מטלה 1

קבוצה 11 A

מגישים:

ירדן חן 305509069

אביב גואטה 305635187

עודד רויטבלט 302222294

אור-אלי שחר 305077901

בר שופר 204341416

תוכן ענייניים:

1. Glossary
2. Functional requirements  
   1. User  
      1. Register to the system.
      2. Login and logout from the system.
      3. Edit user profile which should include an avatar (image of their choice).
      4. Create Texas Hold'em games.
      5. Create game: set game preferences.
      6. Join existing games.
      7. Spectate active game.
      8. Leave a game.
      9. Replay games that are no longer active.
      10. Save favorite turns from replays in order to be viewed on later occasions.
      11. Find all active games which the user can join.
   2. Game Center (System)
3. Store all the information from a game, such as: actions performed by all players in the game, the cards dealt at each round, round beginning and end, etc.
4. List all active games which are available for spectating.
5. Maintain leagues, managing which users are in which league at any given moment.
6. The highest ranking user in the system may set a default league for new users.
7. The highest ranking user in the system may set criteria for moving to a new league.
8. The highest ranking user in the system may move users between leagues.
9. Search/ filter active games by: player name/ pot size/ game preference.
10. Game  
    * 1. Support playing a Texas Hold'em game: dealing cards.
      2. Support playing a Texas Hold'em game: placing blind bets for players.
      3. Support playing a Texas Hold'em game: fold according to the game rules.
      4. Support playing a Texas Hold'em game: allowing players to check (NOP).
      5. Support playing a Texas Hold'em game: call according to the game rules.
      6. Support playing a Texas Hold'em game: raise according to the game rules.
11. Class diagram
12. Component diagram

**Glossary**

* + - * + המערכת – כלי לניהול משחקי פוקר אינטראקטיביים מרובי משתמשים.
        + משתמש חבר - אדם אשר נרשם למערכת ופרטיו נשמרו בבסיס הנתונים של המערכת.
        + פרופיל משתמש – פרטי האדם הרשום למערכת המכילים: כינוי, תמונה.
        + דירוג משתמש – ציון המעיד על כישורי המשחק של המשתמש החבר.
        + ליגה – קבוצת משתמשים בעלי ניקוד הנמצא באותו טווח דירוגים.
        + העדפות משחק – פרטים שונים על הליגה אליה משתייך החדר, סכום ה Small blind, סוג המשחק, כמות כסף מינימלית ומקסימאלית לכניסה.
        + חדר – שולחן פוקר עם העדפות משחק.
        + משחק – משחק פוקר יחיד בחדר (מתחילת המשחק ועד הכרזת המנצח).
        + שחקן – משתמש חבר שנמצא בחדר.
        + צופה – משתמש שנכנס לחדר משחק ויכול להסתכל על המהלכים בו מבלי להשתתף במשחק כשחקן.
        + דילר – השחקן שבמשחק הנוכחי קיבל את סימון ה D.
        + Small blind – סכום ההימור הראשוני של השחקן הבא (עם כיון השעון) לאחר הדילר.
        + Big blind - סכום ההימור הראשוני של השחקן הבא (עם כיון השעון) לאחר ה Small blind – הסכום כפול מערך ה Small blind.
        + חבילת קלפים – חבילת קלפים רגילה המכילה 52 קלפים – 13 מכל אחת מ - 4 הסדרות.
        + ג'יטון – יחידת כסף משחק.
        + התראה – הודעה הנשלחת ע"י המערכת למשתמש/ים או שחקן/ים.

**Register to the system**

ניתוח דרישות \ הנחות:

1. שם משתמש הינו ייחודי.

רישום משתמש למערכת*:*

**:Use case**

1. שם: רישום משתמש חדש למערכת.  
   1.1 תיאור: מאפשר לאדם להירשם למערכת על מנת ליהנות ממגוון השירותים שהיא מציעה.
2. משתתפים: עיקרי – משתמש חדש, משני – מערכת.
3. תנאי קדם: המשתמש אינו מחובר למערכת.
4. תנאי בתר: נוצר משתמש חבר חדש ופרטיו נכנסו למסד הנתונים של המערכת.
5. מהלך האירועים:
   1. מהלך בסיסי:
      1. המשתמש מבקש להירשם למערכת.
      2. המערכת מבקשת מהמשתמש להזין את פרטי הרישום.
      3. **הזנת פרטים**

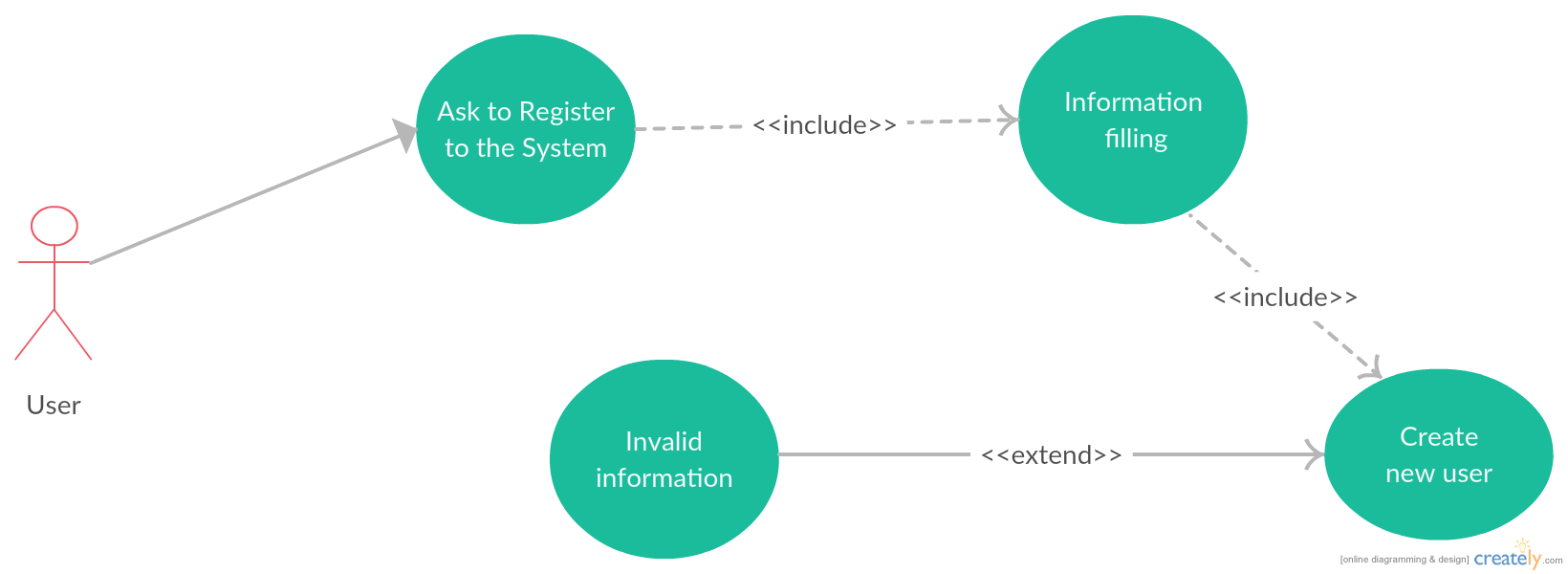
המשתמש מזין את הפרטים המבוקשים.

* + 1. המערכת יוצרת משתמש חבר חדש.
  1. מהלך אלטרנטיבי:
     1. ביטול:  
        המשתמש יכול לבטל את הרישום בכל שלב. במקרה זה לא נוצר משתמש רשום חדש.
     2. פרטים חסרים או שגויים

במהלך שלב **הזנת הפרטים,** במידה והמשתמש לא מילא אחד מהפרטים המבוקשים המערכת תודיע על כך ולא תיצור משתמש חבר חדש.

תרחיש בדיקות קבלה:

1. :Use case רישום משתמש חדש למערכת.
2. Good :
   1. רישום משתמש חדש מוצלח:  
      המשתמש מזין שם משתמש שאינו קיים במערכת, ואת שאר הפרטים הנדרשים ומבקש להירשם, המערכת יוצרת משתמש רשום חדש.
   2. ביטול:  
      במשתמש בוחר באפשרות "חזרה", המהלך האלטרנטיבי "ביטול" (5.2.1) מופעל. מקרה זה נבדק בכל שלב של מהלך האירועים.
3. :Sad
   1. שם משתמש קיים:  
      המשתמש מזין שם משתמש הקיים כבר במערכת ואת שאר הפרטים הנדרשים ומבקש להירשם, המערכת תודיע למשתמש על הבעיה ולא תיצור משתמש חבר חדש.
4. :Bad
   1. קלט לא חוקי:
   2. המשתמש מזין פרטים עם תווים לא חוקיים ושדות ריקים ומבקש להירשם, המערכת תודיע למשתמש על הבעיה ולא תיצור משתמש חבר חדש.



**Login and logout from the system**

התחברות למערכת*:*

**:Use case**

1. שם: התחברות משתמש חבר למערכת.  
   1.1 תיאור: מאפשר למשתמש חבר להתחבר למערכת ולהשתמש בפונקציות המערכת.
2. משתתפים: עיקרי – משתמש חבר, משני – מערכת.
3. תנאי קדם: המשתמש הינו משתמש חבר ואינו מחובר למערכת.
4. תנאי בתר: המשתמש החבר מחובר למערכת.
5. מהלך האירועים:
   1. מהלך בסיסי:
      1. המשתמש החבר מבקש להתחבר למערכת
      2. המערכת מבקשת מהמשתמש להזין את פרטי ההתחברות.
      3. **הזנת פרטים**

המשתמש מזין את פרטיו ומבקש להתחבר למערכת.

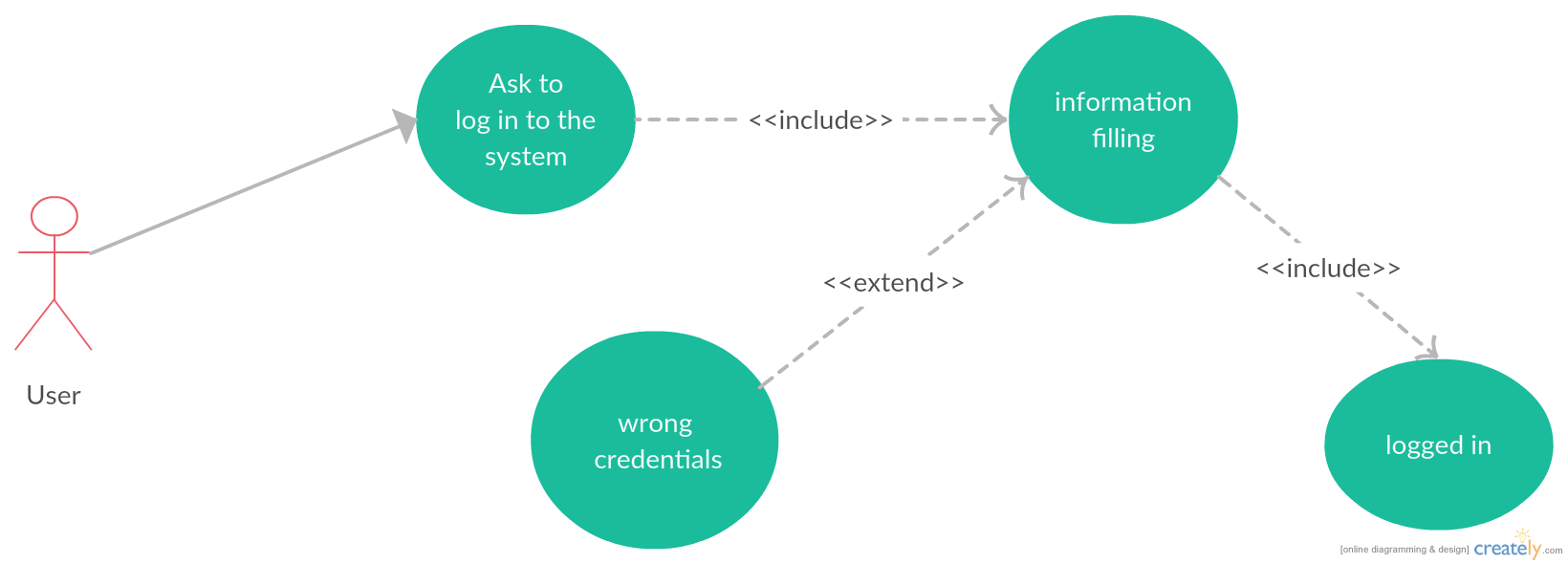
* + 1. המערכת מחברת את המשתמש ומאפשרת לו ליהנות ממגוון הפונקציות שלה.
  1. מהלך אלטרנטיבי:
     1. ביטול:  
        המשתמש יכול לבטל את ההתחברות בכל שלב. במקרה זה המשתמש לא מתחבר למערכת.
     2. פרטים חסרים או שגויים

במהלך שלב **הזנת הפרטים,** במידה והמשתמש לא מילא אחד מהפרטים המבוקשים בהתחברות המערכת תודיע על כך ולא תחבר את המשתמש למערכת.

תרחיש בדיקות קבלה:

1. Use case: התחברות משתמש חבר למערכת.
2. Good:
   1. התחברות משתמש חבר מוצלחת:  
      המשתמש מזין פרטים תקינים ומבקש להתחבר למערכת , המערכת מחברת את המשתמש למערכת.
   2. ביטול:  
      המשתמש מבטל את ההתחברות, המהלך האלטרנטיבי "ביטול" (5.2.1) מופעל. מקרה זה נבדק בכל שלב של מהלך האירועים.
3. Sad:
   1. פרטים שגויים:  
      המשתמש מזין פרטים שגויים ומבקש להתחבר למערכת, המערכת מודיעה למשתמש על הבעיה ולא מתחברת.
4. Bad:
   1. קלט לא חוקי או חסר:

המשתמש מזין פרטים לא חוקיים או חסרים ומבקש להתחבר למערכת, המערכת מודיעה למשתמש על הבעיה ולא מבצעת התחברות.



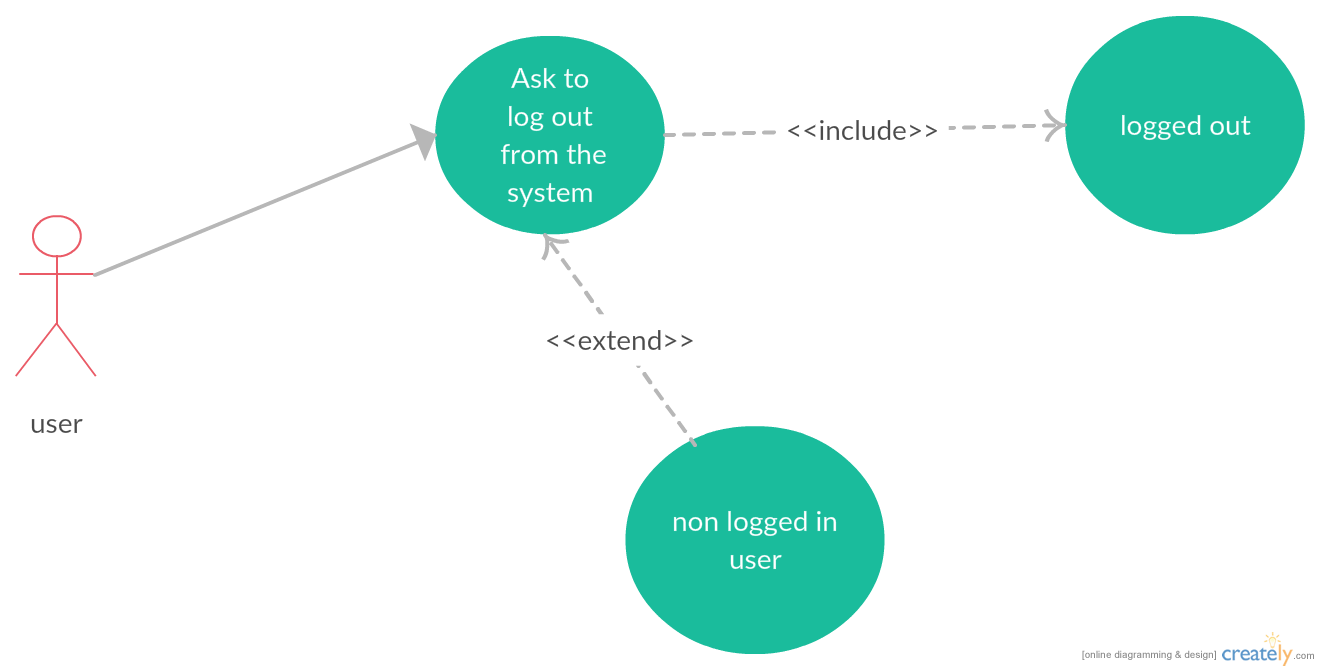
התנתקות מהמערכת*:*

**Use case**:

1. Use case: התנתקות משתמש חבר.  
   1.1 תיאור: מאפשר למשתמש חבר המחובר למערכת, להתנתק ממנה.
2. משתתפים: עיקרי – משתמש חבר, משני – מערכת.
3. תנאי קדם: המשתמש הינו משתמש חבר המחובר למערכת.
4. תנאי בתר: המשתמש החבר התנתק מהמערכת.
5. מהלך האירועים:
   1. מהלך בסיסי:
      1. המשתמש החבר מבקש להתנתק מהמערכת.
      2. המערכת מנתקת את המשתמש מהמערכת.

תרחיש בדיקות קבלה:

1. Use case: התנתקות משתמש חבר.
2. Good:
   1. התנתקות משתמש חבר מחובר:  
      המשתמש מבקש להתנתק מהמערכת , המערכת מנתקת את המשתמש מהמערכת.



**Edit user profile**

ניתוח דרישות \ הנחות**:**

1. שם משתמש הינו ייחודי ולא ניתן לעריכה.
2. כל שאר הפרטי המשתמש ניתנים לעריכה; סיסמא, מייל וכו'.

עריכת פרטי משתמש*:*

**: Use case**

1. שם: עריכת פרטי משתמש.  
   1.1 תיאור: מאפשר למשתמש חבר המחובר למערכת לעדכן את פרטיו.
2. משתתפים: עיקרי – משתמש חבר, משני – מערכת.
3. תנאי קדם: המשתמש מחובר למערכת.
4. תנאי בתר: פרטי המשתמש החבר עודכנו במערכת.
5. מהלך האירועים:
   1. מהלך בסיסי:
      1. המשתמש החבר מבקש לעדכן את פרטיו.
      2. המערכת מבקשת מהמשתמש להזין את הפרטים העדכניים.
      3. **הזנת פרטים**

המשתמש מזין את הפרטים אותם הוא מעוניין לעדכן ומבקש לעדכנם.

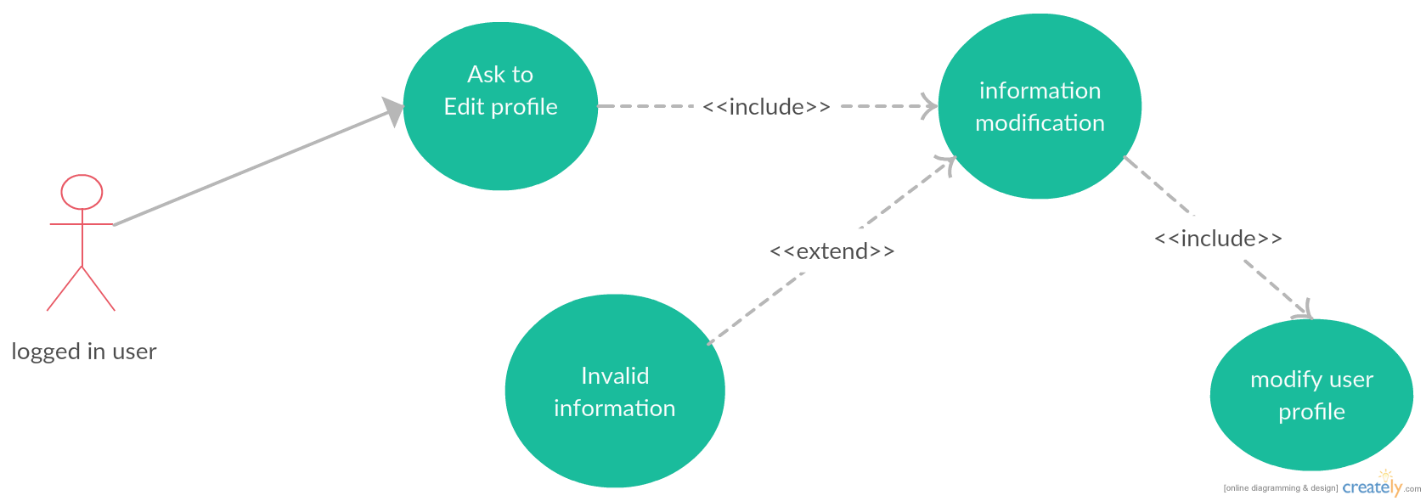
* + 1. המערכת מעדכנת את הפרטים המבוקשים.
  1. מהלך אלטרנטיבי:
     1. ביטול:  
        המשתמש יכול לבטל את עדכון הפרטים טרם סיים לעדכנם. במקרה זה פרטיו לא משתנים.
     2. פרטים שגויים

במהלך שלב **הזנת הפרטים,** במידה והמשתמש הזין תווים או פרטים לא חוקיים המערכת תודיע למשתמש על הבעיה והפרטים של המשתמש לא יעודכנו.

תרחיש בדיקות קבלה:

1. Use case: עריכת פרטי משתמש.
2. Good :
   1. עדכון פרטים מוצלח:  
      המשתמש מזין פרטים חוקיים ומבקש לעדכנם, המערכת מעדכנת את הפרטים.
   2. ביטול:  
      המשתמש בוחר באפשרות ביטול העדכון, המהלך האלטרנטיבי "ביטול" (5.2.1) מופעל.
3. Bad :
   1. קלט לא חוקי:

המשתמש מזין פרטים לא חוקיים ומבקש לעדכנם, המערכת מודיעה למשתמש על הבעיה והפרטים לא מתעדכנים.



**Create Texas Hold'em games**

*יצירת משחק חדש:*

**Use case:**

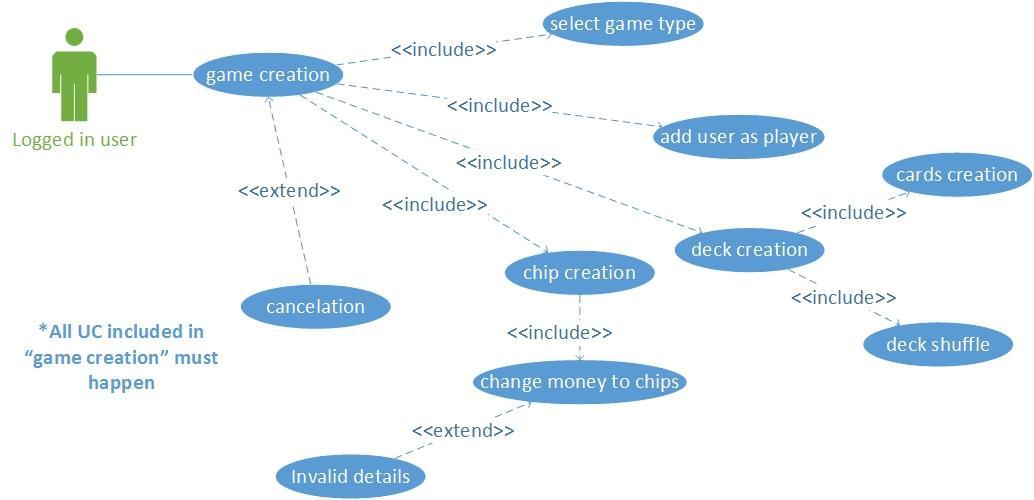
1. Use case: יצירת משחק חדש.  
   1.1 תיאור: מאפשר למשתמשים ליצור חדר משחק חדש. המשתמש בוחר העדפת חדר והמערכת יוצרת חדר משחק חדש אשר השחקן היחיד בו ברגע יצירתו הוא המשתמש.
2. משתתפים: עיקרי – משתמש חבר, משני – מערכת המשחקים.
3. תנאי קדם: המשתמש הינו משתמש חבר.
4. תנאי בתר: קיים חדר משחק חדש במערכת. בחדר יש שחקן יחיד שהוא המשתמש שיצר אותו.
5. מהלך האירועים:
   1. מהלך בסיסי:
      1. המשתמש מבקש ליצור משחק חדש.
      2. בחירת סוג המשחק:  
         מפורט בuse case הבא.
      3. בחירת כמות ג'יטונים:  
         המשתמש בוחר את סכום הכסף איתו הוא מעוניין להיכנס לחדר ואותו ימיר לג'יטונים.
      4. יצירת המשחק:  
         המערכת יוצרת חדר משחק חדש מהסוג שנבחר. המשתמש מצורף למשחק כשחקן. המשחק אינו פעיל.
   2. מהלך אלטרנטיבי:
      1. ביטול:  
         המשתמש יכול לבטל את יצירת המשחק בכל שלב. במקרה זה המשתמש לא מצורף לשום חדר, לא נוצר שום מידע חדש במערכת ולא משתנה מידע קיים במערכת.
      2. בחירה מחודשת של כמות ג'יטונים:  
         במידה והמשתמש בחר סכום כסף אשר אינו בתחום שהוגדר בהעדפות המשחק, או אינו בוחר סכום כלל, המערכת תציג למשתמש הודעת שגיאה והמשתמש יישאר במהלך 5.1.3 בחירת ג'יטונים.

תרחיש בדיקות קבלה:

1. Use case: יצירת משחק חדש.
2. Good:
   1. הצלחת יצירת משחק חדש:

המשתמש בחר העדפת חדר ולא בחר בביטול בשום שלב. נוצר חדר חדש עם העדפת חדר כפי שהמשתמש בחר, המשתמש צורף לחדר כשחקן, הוא השחקן היחיד במשחק ואין בו צופים.

1. Sad:
   1. אי בחירת כמות ג'יטונים:  
      המשתמש לא בחר כמות גי'טונים איתה הוא מעוניין להיכנס לחדר, המערכת מציגה הודעת שגיאה והמשתמש נשאר במהלך אירועים 5.1.3.
2. Bad:
   1. בחירת כמות ג'יטונים לא חוקית:  
      המשתמש הכניס תווים אשר אינם ספרות בבחירת כמות הג'יטונים או שביצע בחירה לא חוקית אחרת. המערכת מציגה הודאת שגיאה והמשתמש נשאר במהלך אירועים 5.1.3



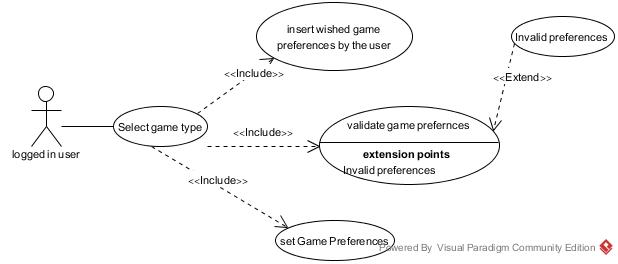
**Create game: set game preferences**

ניתוח דרישות \ הנחות:

1. מאפיין 1: game-type-policy: limit ,no-limit, pot-limit.
2. מאפיין 2: Buy-in-policy: עלות כניסה לחדר משחק.
3. מאפיין 3: chip-policy: כמות הג'יטונים שיש לכל שחקן. במידה והערך המוכנס הינו 0, כל שחקן יישחק עם כל הכסף שברשותו, כלומר, כל הכסף שברשותו יומר לז'יטונים שאיתם ייכנס למשחק.
4. מאפיין 4: ההימור המינימלי (שערכו שווה לbig blind).
5. מאפיין 5: כמות מינימלית ומקסימלית של שחקנים פעילים במשחק ספציפי.
6. מאפיין 6: האם ניתן לאפשר לצופים לצפות במשחק זה או לא.

**Use case:**

1. Use case: הגדרת מאפייני משחק ביצירת משחק חדש.  
   1.1 תיאור: משתמש אשר התחיל פעולת יצירת משחק חדש, בוחר את מאפייני המשחק הרצויים עבורו. המערכת מוודאת שההעדפות שהכניס חוקיים ופעולת יצירת המשחק ממשיכה.
2. משתתפים: עיקרי – משתמש חבר, משני – מערכת המשחקים.
3. תנאי קדם: המשתמש הוא משתמש חבר שהתחיל פעולת יצירת משחק חדש חוקית, והגיע לשלב הגדרת מאפייני המשחק.
4. תנאי בתר: מאפייני המשחק החדש הוגדרו בהצלחה ע"י המשתמש ופעולת יצירת המשחק ממשיכה.
5. מהלך האירועים:
   1. מהלך בסיסי:
      1. המשתמש מזין למערכת את הנתונים לכל מאפייני המשחק.
      2. המערכת מוודאת שהנתונים שהוכנסו חוקיים ותקינים.
      3. המערכת מעדכנת את מאפייני המשחק בהתאם לנתונים שהוכנסו ע"י המשתמש.
   2. מהלך אלטרנטיבי:
      1. הזנת **נתונים לא חוקית. במקרה זה המערכת מתריעה על כך למשתמש ויצירת המשחק מתבטלת**



תרחיש בדיקות קבלה:

1. Use case: הגדרת מאפייני משחק ביצירת משחק חדש.
2. Good:
   1. הצלחת הגדרת מאפייני משחק חדש:

המשתמש בחר העדפת חדר חוקיות. המערכת מעדכנת את העדפות המשחק.

1. Sad:
   1. אי בחירת אחד המאפיינים:  
      המשתמש לא בחר באחד או יותר מהמאפיינים הנדרשים. המערכת מודיעה למשתמש כי המאפיין לא נבחר. שום מידע בערכת לא משתנה, העדפות המשחק לא מתעדכנות.
2. Bad:
   1. תווים לא חוקיים:  
      המשתמש הכניס תווים אשר אינם ספרות או שהינם ספרות שליליות בבחירת אחד או יותר מן המאפיינים 2,3,4,5. המערכת מציגה הודאת שגיאה והמשתמש חוזר למהלך אירועים 5.1.1
   2. בחירת אפסים:  
      המשתמש הכניס ערך 0 בבחירת אחד או יותר מן המאפיינים 4,5. המערכת מציגה הודאת שגיאה והמשתמש חוזר למהלך אירועים 5.1.1

**Join existing games**

ניתוח דרישות \ הנחות:

1. משתמש חבר יכול לצפות ולשחק במשחק.
2. ניתן לבטל את הצטרפות לחדר בכל שלב.
3. משתמש חבר יכול להשתתף כשחקן רק במשחקים התואמים את דירגו ואת הליגה אליה הוא משויך, ויש בהם מקום.
4. ברגע יצירת חדר משחק חדש, משחק מאותחל כ**לא פעיל** ובו השחקן היחיד הינו המשתמש אשר יצר את המשחק.
5. חדר משחק יעבור למצב פעיל רק כאשר יהיו מספיק שחקנים בחדר (2 ומעלה).
6. חדר משחק שאין בו שחקנים נסגר אוטומטית.
7. משתמש לא יכול לצפות ולשחק באותו משחק באותו זמן.
8. משתמש יכול להיות צופה ושחקן בשני משחקים שונים באותו הזמן.

*הצטרפות משתמש למשחק קיים כשחקן:*

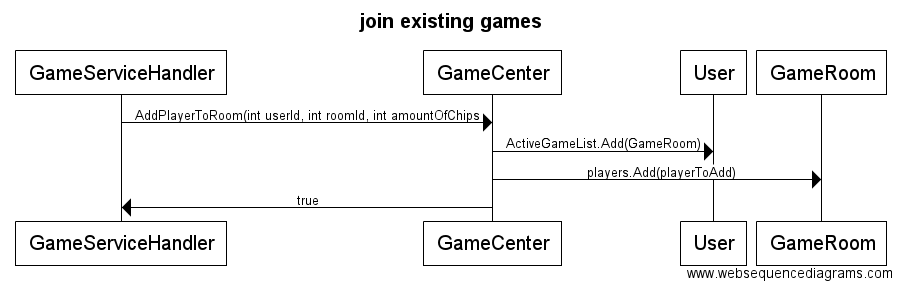
**Use case:**

1. Use case: הצטרפות משתמש למשחק כשחקן.  
   1.1 תיאור: מאפשר למשתמשים להצטרף לחדרים קיימים כשחקנים. משתמשים יוכלו להצטרף לחדר כשחקנים רק בתנאי שהם רשומים ורמת המשחק תואמת לרמתם.
2. משתתפים: עיקרי – משתמש חבר, משני – מערכת המשחקים.
3. תנאי קדם: המשתמש הוא משתמש חבר ומוצגת לו רשימת חדרים אליה הוא רשאי להצטרף כשחקן.
4. תנאי בתר: המשתמש הצטרף למשחק כשחקן.
5. מהלך האירועים:
   1. מהלך בסיסי:
      1. בחירת חדר:  
         המשמש בוחר חדר להשתתף בו כשחקן מתוך רשימת חדרים פעילים אליהם הוא רשאי להצטרף.
      2. המערכת מוודאת כי יש המשתמש רשאי להצטרף למשחק לפי מאפייני המשחק.
      3. בחירת כמות ג'יטונים:  
         המשתמש בוחר את סכום הכסף איתו הוא מעוניין להיכנס לחדר ואותו ימיר לג'יטונים.
      4. צירוף משתמש למשחק:  
         המשתמש מצורף לחדר כשחקן. כסף של המשתמש מומר לג'יטונים.
   2. מהלך אלטרנטיבי:
      1. ביטול:  
         המשתמש יכול לבטל את הצטרפותו בכל שלב. במקרה זה המשתמש לא מצורף לשום חדר ושום חלק מכספו אינו מומר לג'יטונים
      2. בחירה מחודשת של כמות ג'יטונים:  
         במידה והמשתמש בחר סכום כסף אשר אינו בתחום שהוגדר בהעדפות המשחק, המערכת תציג למשתמש הודעת שגיאה והמשתמש יישאר במהלך 5.1.2 בחירת ג'יטונים.
      3. המשתמש אינו רשאי להצטרף למשחק בגלל מאפייני המשחק:  
         במקרה זה המשתמש מקבל התראה על כך ואינו מצטרף למשחק.

תרחיש בדיקות קבלה:

1. Use case: הצטרפות משתמש למשחק כששחקן.
2. Good:
   1. הצלחת הצטרפות כשחקן:  
      המשתמש הינו משתמש חבר והוא בוחר להצטרף לחדר אליו הוא רשאי להצטרף כשחקן, המשתמש מצורף לחדר כשחקן. גודל הכסף של המשתמש יורד. המשתמש מקבל ג’יטונים על פי כמות הכסף שירדה לו.
   2. ביטול:  
      במשתמש בוחר באפשרות ביטול, המהלך האלטרנטיבי "ביטול" (5.2.1) מופעל. מקרה זה נבדק בכל שלב של מהלך האירועים.
3. Sad:
   1. אין חדרים רלוונטיים:  
      לא קיים במערכת אף חדר רלוונטי (פעיל או לא פעיל). לא מוצגים למשתמש חדרים אך ניתן לבצע ביטול.
   2. אי בחירת כמות ג'יטונים:  
      המשתמש לא בחר כמות גי'טונים איתה הוא מעוניין להיכנס לחדר, ובהגדרות החדר כמות הג'יטונים לכניסה גדולה מ-0, המערכת מציגה הודעת שגיאה והמשתמש נשאר במהלך אירועים 5.1.2.
4. Bad:
   1. הכנסת תווים לא חוקיים:  
      המשתמש הכניס תווים אשר אינם ספרות או הינם מספר שלילי בשדה כמות הג'יטונים. המערכת מציגה הודעת שגיאה והמשתמש נשאר במהלך 5.1.2

תרשים רצף:



**Spectate active game**

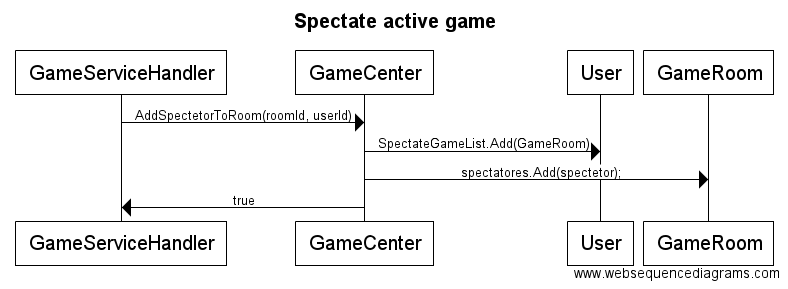
*הצטרפות משתמש לחדר כצופה:*

**Use case:**

1. Use case: הצטרפות משתמש לחדר כצופה.  
   1.1 תיאור: מאפשר למשתמשים להצטרף לחדרים קיימים כצופים.
2. משתתפים: עיקרי – משתמש חבר או לא חבר, משני – מערכת המשחקים.
3. תנאי קדם: מוצגת למשתמש רשימת חדרים פעילים.
4. תנאי בתר: המשתמש הצטרף לחדר כצופה.
5. מהלך האירועים:
   1. מהלך בסיסי:
      1. בחירת חדר:  
         המשמש בוחר חדר לצפות בו מתוך רשימת חדרים פעילים.
      2. המערכת מוודאת כי ניתן לצפות בחדר הנבחר בהתאם למאפייני המשחק.
      3. צירוף משתמש לחדר:  
         המשתמש מצורף לחדר כצופה.
   2. מהלך אלטרנטיבי:
      1. ביטול:  
         המשתמש יכול לבטל את הצטרפותו בכל שלב. במקרה זה המשתמש לא מצורף לשום חדר.
      2. לא ניתן לצפות בחדר בהתאם למאפייני המשחק:  
         במצב זה המשתמש אינו נכנס לחדר כצופה.

תרחיש בדיקות קבלה:

1. Use case: הצטרפות משתמש לחדר כצופה.
2. Good:
   1. הצלחת הצטרפות כצופה:  
      המשתמש בוחר להצטרף כצופה לחדר, המשתמש מצורף לחדר כצופה. כמות הכסף שלו לא השתנתה ואין לו ג’יטונים בחדר זה.
   2. ביטול:  
      במשתמש בוחר באפשרות ביטול, המהלך האלטרנטיבי "ביטול" (5.2.1) מופעל. מקרה זה נבדק בכל שלב של מהלך האירועים.
3. Sad:
   1. אין חדרים פעילים:  
      לא קיים במערכת אף חדר פעיל. לא מוצגים למשתמש חדרים אך ניתן לבצע ביטול.



**Leave a game**

ניתוח דרישות / הנחות:

1. כאשר שחקן יוצא ממשחק המשחק יצטרף לרשימת המשחקים שניתנים לשחזור עבור המשתמש.
2. כאשר שחקן יוצא ממשחק כמות הכסף של המשתמש תתעדכן בהתאם.
3. כאשר משחק מסתיים, דירוג השחקנים שהשתתפו בו יתעדכן.
4. קביעת דירוג המשתמשים תקבע ע"י הנוסחא:
5. *משתמש שלא היה פעיל חודש ומעלה יורד ליגה אוטומטית.*
6. כאשר שחקן יוצא ממשחק, אם כמות השחקנים שנשארו בו קטנה מ- 2 אזי המשחק יסתיים והחדר יעבור למצב לא פעיל.
7. כאשר שחקן יוצא ממשחק פעיל טרם סיומו, הדבר נחשב לו כהפסד.
8. אם שחקן שהיה מוגדר כדילר יוצא מהמשחק, הדילר יוגדר להיות השחקן הבא עם כיוון השעון.
9. אם שחקן שהיה מוגדר כSB יוצא מהמשחק, הSB יוגדר להיות השחקן הבא עם כיוון השעון.
10. אם שחקן שהיה מוגדר כBB יוצא מהמשחק, הBB יוגדר להיות השחקן הבא עם כיוון השעון.
11. כאשר שחקן יוצא מהמשחק השחקנים שנשארו בחדר יקבלו הודעה על כך שהמשתמש הנ"ל יצא מהמשחק.

דרישות לא פונקציונאליות:

1. יציאה מהמשחק ושמירת הלוג לצורכי שחזור עבור משתמש תהיה מהירה.

שאלות ללקוח:

1. אם שחקן יוצא בסבב האחרון לפני שהציגו את הקלפים והוא ניצח האם הוא מקבל את הניצחון והכסף או שהניצחון עובר למקום השני?

תשובה שלנו: המקום השני מנצח.

1. אם שחקן יוצא בסבב האחרון אחרי שהציגו את הקלפים ולפני הכרזת המנצח וכשהוא עם היד החזקה האם הוא מקבל את הניצחון והכסף או שהניצחון עובר למקום השני?

תשובה שלנו: המשתמש עדין מנצח.

*יציאת משתמש ממשחק:*

**Use case:**

1. Use case: יציאת משתמש ממשחק.  
   1.1 תיאור: משתמש אשר נמצא במשחק מבקש לצאת ממנו לגמרי (לא לבצע logout).
2. משתתפים: עיקרי -המשתמש אשר רוצה לצאת מהמשחק, משני - והשחקנים אשר נמצאים במשחק ממנו הוא פורש.
3. תנאי קדם: המשתמש מחובר למערכת. המשחק פעיל. והמשתמש נמצא בחדר המשחק או כצופה או כשחקן פעיל.
4. תנאי בתר: , הכסף של המשתמש התעדכן בהתאם, דרגת המשתמש עודכנה בהתאם.
5. מהלך האירועים:
   1. מהלך בסיסי:
      1. בקשת יציאה מהמשחק:  
         המשתמש מבקש לצאת מהמשחק.
      2. שמירת Game log:  
         המערכת שומרת את פרטי המשחק תחת log game עבור המשתמש לצורכי שחזור עתידיים.
      3. עדכון כסף:  
         המערכת מעדכנת את הכסף של המשתמש בארנק.
      4. עדכון דרגת שחקן:  
         המערכת מעדכנת את דרגת המשתמש.
      5. במשחק מעל 2 שחקנים:

קיימים יותר מ2 שחקנים בחדר – המערכת מודיעה לשחקנים שנותרו בחדר כי המשתמש הנ"ל יצא ובמידה והוא היה הדילר / SB / BB מעדכנת תפקידים בהתאם.

* + 1. יציאה מהמשחק:

המערכת תוציא את השחקן שביקש לצאת מחדר המשחק.

* 1. מהלך אלטרנטיבי:
     1. במשחק 2 שחקנים (כולל השחקן שפרש):  
        קיימים 2 שחקנים בחדר (כולל המשתמש שלנו) – המערכת מודיעה לשחקן שנותר בחר כי המשתמש הנ"ל יצא מהמשחק וכי המשחק נסגר.

תרחיש בדיקות קבלה:

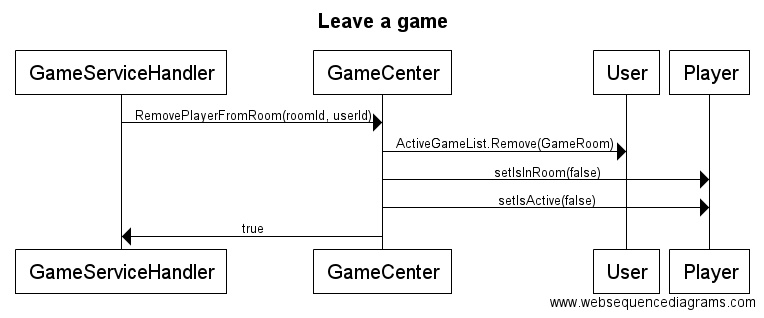
1. use case: המשתמש מבקש לצאת ממשחק

2. Good:

2.1 יציאה מוצלחת מהמשחק:

המשתמש שביקש לצאת מהחדר, יוצא בהצלחה, הכסף שזכה / הפסיד התעדכן בארנק אוטומטית וכך גם הדירוג שלו.

תרשים רצף:



**Replay games that are no longer active**

ניתוח דרישות / הנחות:

1. הצפייה תהיה באותה מהירות בה המשחק התקיים (פרט לזמני התגובה של השחקנים).
2. בזמן הצגת השחזור יוכל המשתמש לעצור את המשחק ולצאת מהמשחק.
3. בזמן השחזור יוכל השחקן לבחור מהלך מועדף לצפייה מועדפת.
4. רק שחקנים שמחוברים יכולים לצפות יכולים לצפות במשחקים שהתרחשו בעבר.
5. שחקן יכול לצפות רק במשחקים שהשתתף בהם.
6. לשחקן יש את רשימת המשחקים והחדרים שבהם הוא יכול לצפות.

תרחיש בדיקות קבלה:

1. Use case: המשתמש מבקש לראות את כל המשחקים הלא פעילים אותם ניתן לשחזר.
2. Good:
   1. הצגה מוצלחת:

המערכת מציגה למשתמש את רשימת המשחקים הלא פעילים בהם השתתף הניתנים לשחזור.

1. Sad:
   1. הצגה ריקה:

המערכת מציגה למשתמש רשימה ריקה של משחקים לא פעילים הניתנים לשחזור, לא קיימים משחקים לא פעילים עבור משתמש זה.

**תרחיש בדיקות קבלה נוסף:**

1. Use case: המשתמש בוחר משחק לשחזור מתוך רשימת המשחקים לשחזור.
2. Good:
   1. שחזור מוצלח:

המערכת משחזרת בהצלחה את המשחק אותו בחר המשתמש להצגה בהצלחה ומציגה אותו למשתמש.

הצגת שחזור של משחק לא פעיל:

**Use case:**

1. Use case: שחזור משחק ע"י משתמש של משחק לא פעיל.  
   1.1 תיאור: משתמש מבקש לראות שחזור של משחק לא פעיל
2. משתתפים: עיקרי - המשתמש אשר רוצה לצפות בשחזור המשחק
3. תנאי קדם: המשתמש מחובר למערכת. המשחק המבוקש בו רוצה המשתמש לצפות כבר לא פעיל.
4. תנאי בתר: המערכת משחזרת בהצלחה את המשחק המבוקש לשחזור ומציגה אותו למשתמש.
5. מהלך האירועים:
   1. מהלך בסיסי:
      1. בקשת רשימת שחזורים:  
         המשתמש מבקש מהמערכת את הרשימה של המשחקים אותו הוא יכול לשחזר.
      2. הצגת משחקים:

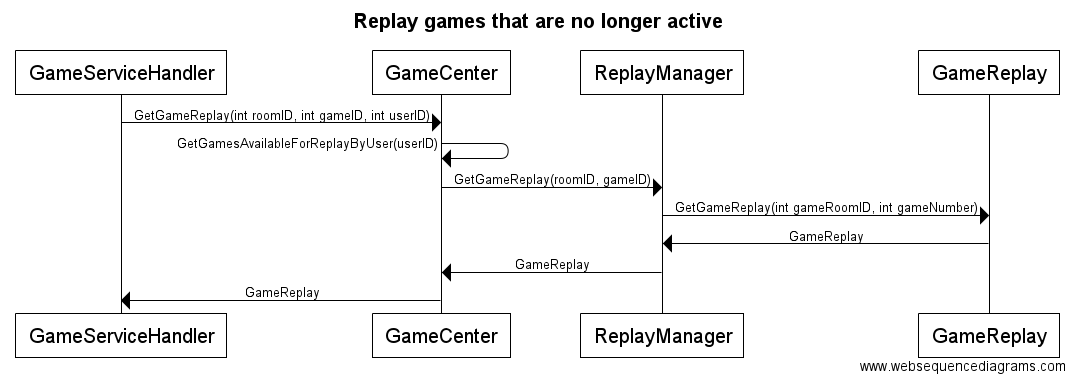
המערכת מציגה למשתמש את המשחקים הנתינים לשחזור שכבר לא פעילים.

* + 1. בחירת משחק לשחזור:   
       המשתמש בוחר המשחק לשחזור
    2. צפייה בשחזור ללא הפסקה:  
       המשתמש צופה בשחזור המשחק מתחילתו ועד הסוף.
    3. סיום השחזור:

המערכת תסיים את שחזור המשחק.

* 1. מהלך אלטרנטיבי:
     1. הפסקת הצפייה בשחזור באמצע:  
        המשתמש בוחר לצאת מהשחזור באמצע

תרשים רצף: שחזור משחק לא פעיל



**DELETE**

**Save favorite turns from replays in order to be viewed on later occasions**

ניתוח דרישות / הנחות:

1. לא יהיה ניתן לשמור את אותו המהלך פעמים.
2. המהלכים ישמרו ברשימה של כל המהלכים המועדפים ע"י השחקן.
3. במהלך השחזור בכל מהלך שבוצע ע"י השחקן תהיה אופציה לשמירת "מהלך אחרון".
   1. אם נבחר מהלך ששמור כבר במהלכים המועדפים, השמירה לא תתבצע והמשתמש יקבל הודעת חיווי ששמירת המהלך לא נשמרה עקב כך שהמהלך הנ"ל כבר שמור.
   2. אם נבחר מהלך שטרם נשמר במהלכים מועדפים, המהלך ישמר תחת המהלכים המועדפים,והמשתמש יקבל הודעת חיווי על שמירה מוצלחת.

דרישות לא פונקציונאליות:

1. השמירה צריכה להיות קלה ואינטואיטיבית למשתמש.

הנחות ושאלות ללקוח:

1. האם שחקן יכול לשמור רק את המהלכים המועדפים שלו? או ם של שחקנים אחרים?

תשובה שלנו: לא, רק את של עצמם.

1. האם מהלך נחשב כ call / raise / fold. או בסבב יחיד?

תשובה שלנו: מהלך כ call / raise / fold

תרחיש בדיקות קבלה:

1. Use case: בחירת מהלך לשמירה מתוך שחזור
2. Good:
   1. שמירה מוצלחת:

המשתמש בוחר מהלך שטרם נשמר במערכת, המערכת שומרת את המהלך בהצלחה ומוסיפה אותו לרשימת המהלכים המועדפים ומודיע בהודעת חיווי למשתמש על ההצלחה.

1. Sad:
   1. המשתמש בוחר מהלך לשמירה למהלכים מועדפים, המהלך כבר שמור במערכת ולכן המהלך לא נשמר בהצלחה במערכת.

שמירת מהלך מועדף מתוך שחזור משחק:

**Use** **case**:

1. Use case: שמירת מהלך מעודף מתוך שחזור במטרה לצפייה חוזרת במועדים אחרים  
   1.1 תיאור: משתמש מבקש לשמור מהלך מתוך שחזור ולהוסיף אותו למהלכים מועדפים.
2. משתתפים: עיקרי - המשתמש אשר צופה בשחזור ובוחר את המהלך.
3. תנאי קדם: המשתמש מחובר למערכת. המשתמש צופה בשחזור משחק לא פעיל. המהלך אותו הוא רוצה לשמור למועדפים טרם נשמר.
4. תנאי בתר: המערכת שומרת את המהלך בהצלחה ומוסיפה אותו לרשימת המהלכים המועדפים.
5. מהלך האירועים:
   1. מהלך בסיסי:
      1. בחירת מהלך:

המשתמש בוחר מהלך לשמירה.

* + 1. בדיקה האם כבר נשמר:

המערכת בודקת האם המהלך נשמר בעבר ברשימת המהלכים המועדפים.

* + 1. שמירת המהלך:

המערכת שומרת את המהלך הנבחר ומוסיפה אותו לרשימת המהלכים המועדפים.

* + 1. הודעת חיווי על הצלחה:

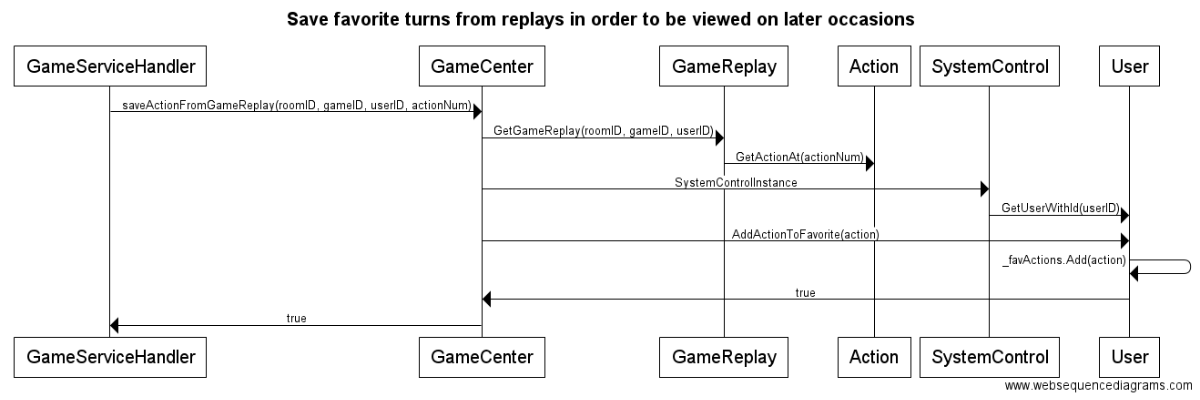
המערכת מודיעה למשתמש ע"י הודעת חיווי למשתמש על הצלחת שמירת המהלך.

* 1. מהלך אלטרנטיבי:
     1. המהלך כבר שמור:  
        המהלך הנבחר כבר שמור – לכן המערכת לא שומרת את המהלך.

הודעת חיווי על כישלון:

המערכת מודיעה למשתמש הודעת חיווי על כך שהמהלך כבר נשמר בעבר וקיים במהלכים מועדפים.

תרשים רצף: שמירת מהלך מועדף



**Find all active games which the user can join**

ניתוח דרישות \ הנחות:

1. משתמש חבר שזה עתה נירשם למערכת מקבל דירוג בסיסי.
2. משתמש חבר יכול לצפות ולשחק במשחק.
3. המערכת תשמור לכל משתמש דירוג שיתעדכן בכל פעם שהמשתמש יסיים לשחק במשחק.
4. משתמש חבר יכול להשתתף כשחקן רק במשחקים התואמים את דירגו ואת הליגה אליה הוא משויך, ויש בהם מקום.
5. ניתן לבטל את ההצטרפות לחדר על ידי לחיצה על מקש ביטול בכל שלב.
6. חדר משחק יעבור למצב פעיל רק כאשר יהיו מספיק שחקנים בחדר (2 ומעלה).
7. משתמש לא יכול לצפות ולשחק באותו משחק באותו זמן.
8. משתמש יכול להיות צופה ושחקן בשני משחקים שונים באותו הזמן.

חיפוש כלל המשחקים הפעילים אליהם המשתמש יכול להצטרף*:*

**Use case:**

:use case .1

מציאת כל המשחקים הפעילים, אליהם יכול המשתמש חבר להצטרף.

1.1 תיאור:

מציאת משחקים למשתמשים חבר על ידי המערכת על מנת שיוכלו להצטרף לחדרים קיימים כשחקנים. משתמשים חברים יוכלו להצטרף לחדר כשחקנים רק בתנאי שהם רשומים ורמת המשחק תואמת לרמתם.

2. משתתפים: עקרי- משתמש חבר, משני- המערכת.

3. תנאי קדם: המשתמש הוא משתמש חבר ומחובר למערכת.

4. תנאי בתר: המערכת מציגה למשתמש את רשימת חדרים אליה הוא רשאי להצטרף כשחקן.

5. מהלך האירועים:

5.1 מהלך בסיסי:

5.1.1 המשתמש מבקש לבדוק לאילו חדרים הוא רשאי להצטרף.

5.1.2 המערכת בודקת בהתאם לדירוג המשתמש ובהתאם לחדרים שפעילים כעת במערכת לאילו מן החדרים הוא רשאי להצטרף ומציגה אותם למשתמש.

5.1.3 לאחר שמוצגת בפני המשתמש רשימת החדרים, הוא יכול לבחור לאילו מן החדרים הוא רוצה להצטרף.

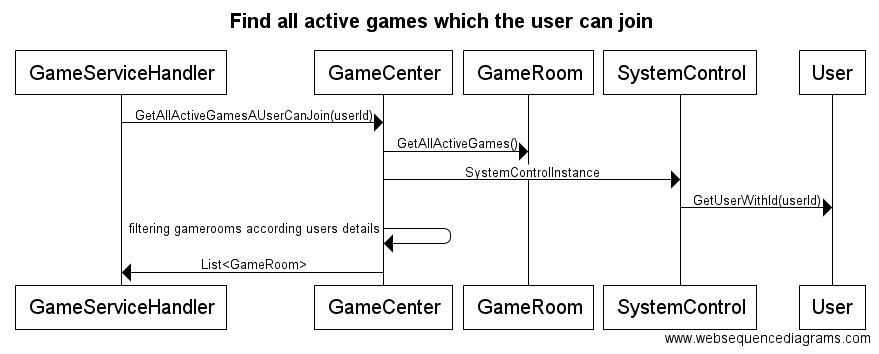
5.1.3.1 המשתמש בוחר להצטרף לחדר כשחקן.

5.2 מהלך אלטרנטיבי:

    5.2.1 המשתמש מחליט לבטל את החיפוש.

תרחיש בדיקות קבלה:

1. Use case: המשתמש בודק לאיזה משחקים הוא רשאי להצטרף.
2. Good:
   1. החיפוש הסתיים בהצלחה:  
      המשתמש הוא משתמש חבר, המערכת מציגה למשתמש את כלל החדרים שתואמים לדירוגו והוא יכול להצטרף אליהם. עתה, יכול המשתמש לבחור לאיזה חדר הוא מעוניין להצטרף.
   2. ביטול:  
      המשתמש בוחר באפשרות ביטול, המהלך האלטרנטיבי "ביטול" (5.2.1) מופעל.
3. Sad:
   1. המערכת מציגה למשתמש רשימה ריקה, אין חדרים פעילים:  
      לא קיים במערכת אף חדר פעיל התואם לדירוגו של המשתמש.



**Store all the information from a game, such as: actions performed by all players in the game, the cards dealt at each round, round beginning and end, etc.**

ניתוח דרישות \ הנחות:

1. כל חדר יידרש בניהול logger
2. הlogger יתועדו כלל הפעילויות שקורות בחדר.
3. תהיה מחלקת logger ראשית שבעת סיום המשחק בחדר תקבל את הlogger.
4. כל חדר / אובייקט logger יקבל מזהה חד ערכי הייחודי לו.

אחסון כלל המידע של המשחקים - פעולות המבוצעות על ידי השחקנים במשחק, הקלפים שחולקו, מהלכים בסיסיים הקורים בו, ועוד *:*

**Use case:**

1. use case: אחסון כלל המידע של חדרי המשחקים - פעולות המבוצעות על ידי השחקנים במשחק, הקלפים שחולקו, מהלכים בסיסיים הקורים בו, ועוד.

1.1 תיאור:

עם תחילת המשחק בחדר, מתבצע איֿתחול לאובייקט פנימי של logger. בעת ביצוע כל action במשחק, תוכנס שורת תיעוד לגבי כך באובייקט הlogger. עם סיום המשחק בחדר ישלח אובייקט הlogger יחד עם מספר מזהה של החדר למערכת המשחקים.

2. משתתפים: לא רלוונטי.

3. תנאי קדם: אובייקט log מאותחל. האובייקט/ חדר בעלי מספר מזהה ייחודי.

4. תנאי בתר: בסיום המשחק בחדר נשלח הlog יחד עם מספר מזהה למערכת המשחקים הראשית.

5. מהלך האירועים:

5.1 מהלך בסיסי:

5.1.1 עם יצירת חדר חדש , החדר מקבל מס׳ מזהה ייחודי.

5.1.2. יצירת החדר מאתחלת מופע של logger, שמקבל מס׳ מזהה גם כן.

5.1.3 כל פעולה שמתבצעת במשחק מתועדת לlogger.

5.1.4 בסיום המשחק בחדר הlogger מועבר למערכת המשחקים.

5.2 מהלך אלטרנטיבי:

5.2.1 מס׳ מזהה שאינו ייחודי לחדר: מערכת המשחקים מקבלת קובץ logger  יחד עם

מס׳ מזהה של חדר שכבר קיים. ולכן מתבצעת דריסה של קובץ

logger של חדר אחר.

5.2.2 תיעוד הלוקה בחסר: המשחק לא מתועד כמו שצריך, למשל, בכל פעם שקורת

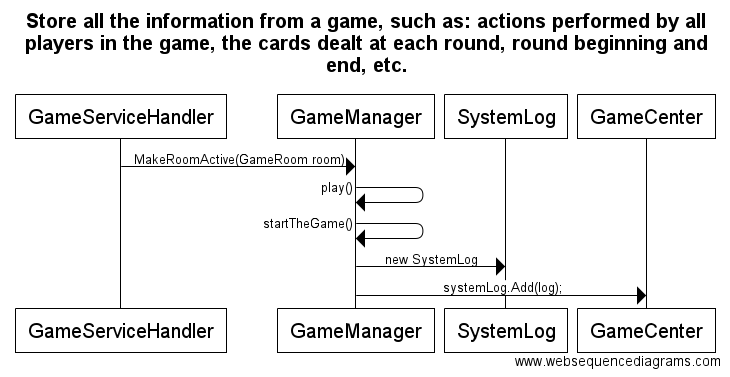
פעולה בסיסית כמו מעבר לסיבוב חדש, אינה מתועדת לקובץ הlogger.

**תרחיש בדיקות קבלה:**

1. Use case:

 אחסון כלל המידע של החדרים - פעולות המבוצעות על ידי השחקנים במשחק, הקלפים שחולקו, מהלכים בסיסיים בו, ועוד.

1. Good:
   1. שמירה מוצלחת:  
      משחק מסתיים, המערכת שומרת את כלל המידע הרלוונטי אליו ב-log.



**List all active games which are available for spectating.**

**ניתוח דרישות \ הנחות**:

1. על מנת שמשתמש יוכל לצפות במשחק, עליו להתחבר למערכת.
2. בכדי לצפות במשחק על החדר להיות פעיל.
3. מערכת המשחק תחזיק רשימה של כלל החדרים הפעילים שניתן להצטרף אליהם כצופים.
4. המשתמש יכול להשתתף בצ׳אט בחדר המשחק ולהגיב בו באופו פעיל כמו שחקן פעיל במשחק.
5. הנחה: משתמש יוכל להתחבר לצפות במשחק, גם אם לא תואם את דירוגו.
6. הנחה: אין הגבלת מס׳ צופים במשחק.
7. הנחה: בכל שלב של במשחק ניתן להצטרף אליו כצופה.

**הצגת כל המשחקיים הקיימים שניתן להצטרף אליהם כצופה*:***

Use case:

המערכת תציג למשתמשים חדרים פעילים שניתן להצטרף אליהם כצופים

:Use case .1

1.1 תיאור:

משתמש מתחבר למערכת ומבקשת לראות את כלל החדרים הפעילים כעת במערכת. (הנחה: גם כאלה שלא תואמים את הדירוג שלו). המערכת לא תבדוק את דירוג המשתמש, ותייצר פלט המכיל את כלל החדרים הפעילים כעת.

2. משתתפים: עיקרי – משתמש חבר או לא חבר, משני – מערכת המשחקים

3. תנאי קדם: ישנם חדרים פעילים. המשתמש מחובר למערכת.

4. תנאי בתר: מוצגת למשתמש רשימת חדרים פעילים.

5. מהלך האירועים:

5.1 מהלך בסיסי:

5.1.1 המשתמש מבקש לבדוק אילו חדרים פעילים כעת במערכת.

5.1.2 המערכת בודקת אילו חדרים פעילים וניתנים לצפייה כעת, ומציגה רשימה שלהם למשתמש.

5.1.3 לאחר שמוצגת בפני המשתמש רשימת חדרים, הוא בוחר לאילו מהם הוא רוצה

             להצטרף כצופה.

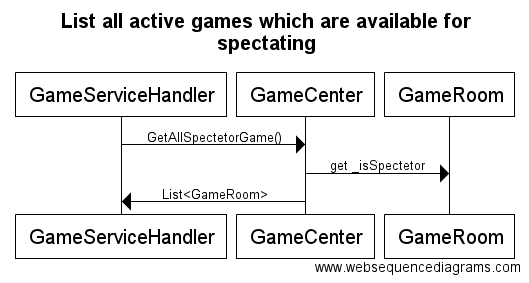
5.1.4 המשתמש מצטרף לחדר ע״י המערכת.

5.2 מהלך אלטרנטיבי:

5.2.1 המשתמש מחליט לבטל את החיפוש.

**תרחיש בדיקות קבלה**:

1. Use case: הצגת כל החדרים הפעילים שניתן להצטרף אליהם כצופה.
2. Good:
   1. החיפוש הסתיים בהצלחה:  
      המשתמש הוא משתמש חבר, המערכת מציגה למשתמש את כלל החדרים הפעילים כעת במערכת ושהוא רשאי להצטרף אליהם כצופה. עתה יכול המשתמש לבחור לאיזה חדר הוא מעוניין להצטרף.
   2. החיפוש הופסק:  
      המשתמש בוחר באפשרות ביטול, המהלך האלטרנטיבי "ביטול" (5.2.1) מופעל.
3. Sad:
   1. המערכת מציגה למשתמש רשימה ריקה, משום שאין חדרים פעילים הניתנים לצפייה כעת במערכת:



**Maintain leagues, managing which users are in which league at any given moment**

**ניתוח דרישות \ הנחות:**

1. מערכת המשחק בודקת את מצב הפעילות של כל שחקן בכניסתו למערכת, ומעדכנת את מצבו (אם צריך).
2. מס' הליגות הוא סופי, ומוגדר מראש במערכת. כמו כן, מוגדרות הדרישות משחקן על מנת להיות ב-ליגה ספציפי.
3. משתמש חדש באופן דיפולטי בליגה ההתחלתית (הנמוכה ביותר) באופן אוטומטי.
4. מערכת המשחק תחזיק נתונים של כלל השחקנים והליגות שהם נמצאים בהן. נתונים אלה יתעדכנו בכל שינוי.
5. מערכת המשחק תהיה אחראית לזהות מצב בו משתמש עומד בדרישות הנדרשות כדי להעלות\לרדת ליגה, ותהיה אחראית לבצע את השינוי בליגה בעת הצורך.

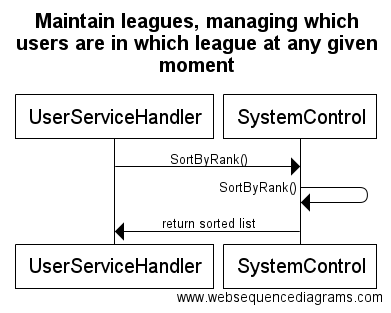
Use case:

1. ניהול ליגות ע"י המערכת והחזרת ליגה נוכחית למשתמש לבקשתו.
   1. תיאור: שחקן במערכת מבקש ממערכת המשחק את טבלת הדירוגי הליגות, המערכת שולפת את הנתונים ומראה למשתמש את המידע ואת מיקומו בה.
2. משתתפים: עקרי - מערכת המשחק, משני - שחקן.
3. תנאי קדם: המשתמש הוא משתמש חבר במערכת.
4. תנאי בתר: מערכת המשחק מראה למשתמש את רשימת הליגות במערכת.
5. מהלך האירועים:
   1. מהלך בסיסי:
      1. המשתמש מבקש מהמערכת את טבלת דירוגי הליגות.
      2. המערכת שולפת את הנתונים.
      3. המערכת בודקת שהנתונים מעודכנים.
      4. המערכת מחזירה למשתמש את טבלת דירוגי הליגות.

**תרחיש בדיקות קבלה:**

1. Use case: מערכת המשחק שולפת את טבלת דירוגי הליגות הנוכחית והמעודכנת , ומחזירה אותה למשתמש.
2. Good:
   1. הצגה מוצלחת:  
      המערכת מחזירה למשתמש את טבלת דירוגי הליגות הנוכחית והמעודכנת

תרשים רצף:

****

**DELETE**

**The highest ranking user in the system may set a default league for new users**

**ניתוח דרישות \ הנחות:**1) ליגת שחקן מאופיינת בטווח דירוגים (rank), לכן, עבור קביעת ליגת ברירת מחדל  
 למשתמשים חדשים בעצם ייקבע דירוג ברירת המחדל ההתחלתי לכל משתמש חדש .  
2) רק המשתמש בעל ה-Rank הגבוה ביותר רשאי לקבוע ליגת ברירת מחדל למשתמשים  
 חדשים.

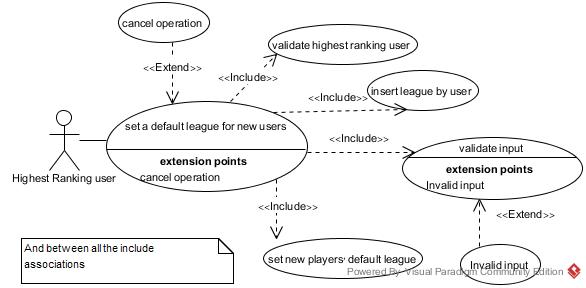
**Use Case:**

1. שם: קביעת ליגת ברירת מחדל עבור משתמשים חדשים.  
   1.1. תיאור: המשתמש בעל הדירוג הגבוה ביותר קובע את ליגת ברירת המחדל עבור כל המשתמשים החדשים המערכת המשחק.
2. משתתפים: עקרי - שחקן, משני - מערכת המשחק.
3. תנאי קדם: המשתמש המבקש לבצע את הפעולה מחובר למערכת באופן תקין וחוקי והוא המשתמש בעל הדירוג הגבוהה ביותר במערכת.
4. תנאי בתר: נקבעת ליגת ברירת מחדל עבור כל המשתמשים החדשים.
5. מהלך האירועים:
   1. מהלך בסיסי:  
      5.1.1. המשתמש מבקש לקבוע ליגת ברירת מחדל עבור כל המשתמשים החדשים.

5.1.2. המערכת בודקת האם המשתמש הוא המשתמש בעל הדירוג הגבוה  
 ביותר.  
 5.1.3. המערכת מבקשת מהמשתמש להזין את הקריטריון לליגת ברירת  
 המחדל.  
 5.1.4. המשתמש מזין את הקריטריון.  
 5.1.5. המערכת בודקת אם הקריטריון שהוזן חוקי ותקין.  
 5.1.6. המערכת מאשרת את הקריטריון החדש, וקובעת אותו כקריטריון   
 לליגת ברירת המחדל.

5.2. מהלך אלטרנטיבי:

5.2.1. ביטול: המשתמש יכול לבטל את קביעת הקריטריון לליגת ברירת  
 המחדל טרם סיים לעדכנם. במקרה זה הקריטריון למעבר לליגה חדשה  
 לא נקבע.  
 5.2.2. הזנת קריטריון שגוי: במהלך שלב הזנת הקריטריון, המערכת מזהה  
 כי הקריטריון שהוזן שגוי (מבחינת Syntax או מבחינת חוקיות לוגית   
 של המשחק). במקרה זה המערכת תודיע למשתמש והקריטריון לא  
 ייקבע.



**DELETE**

**The highest ranking user in the system may set criteria for moving to a new league**

**ניתוח דרישות \ הנחות:**1) מעבר לליגה חדשה מתבצע אך ורק בעקבות שינוי ב-Rank.  
2) רק המשתמש בעל ה-Rank הגבוה ביותר רשאי לקבוע קריטריון למעבר לליגה חדשה.

**Use Case:**

1. שם: קביעת קריטריון למעבר לליגה חדשה.  
   1.1. תיאור: המשתמש בעל הדירוג הגבוה ביותר קובע את הקריטריון למעבר לליגה חדשה.
2. משתתפים: עקרי - שחקן, משני - מערכת המשחק.
3. תנאי קדם: המשתמש המבקש לבצע את הפעולה מחובר למערכת באופן תקין וחוקי והוא המשתמש בעל הדירוג הגבוהה ביותר במערכת.
4. תנאי בתר: נקבע קריטריון למעבר לליגה חדשה.
5. מהלך האירועים:
   1. מהלך בסיסי:  
      5.1.1. המשתמש מבקש לקבוע קריטריון לליגה חדשה.

5.1.2. המערכת בודקת האם המשתמש הוא המשתמש בעל הדירוג הגבוה  
 ביותר.  
 5.1.3. המערכת מבקשת מהמשתמש להזין את הקריטריון לליגה חדשה.  
 5.1.4. המשתמש מזין את הקריטריון.  
 5.1.5. המערכת בודקת אם הקריטריון שהוזן חוקי ותקין.  
 5.1.6. המערכת מאשרת את הקריטריון החדש, וקובעת אותו כקריטריון   
 למעבר לליגה חדשה.

5.2. מהלך אלטרנטיבי:

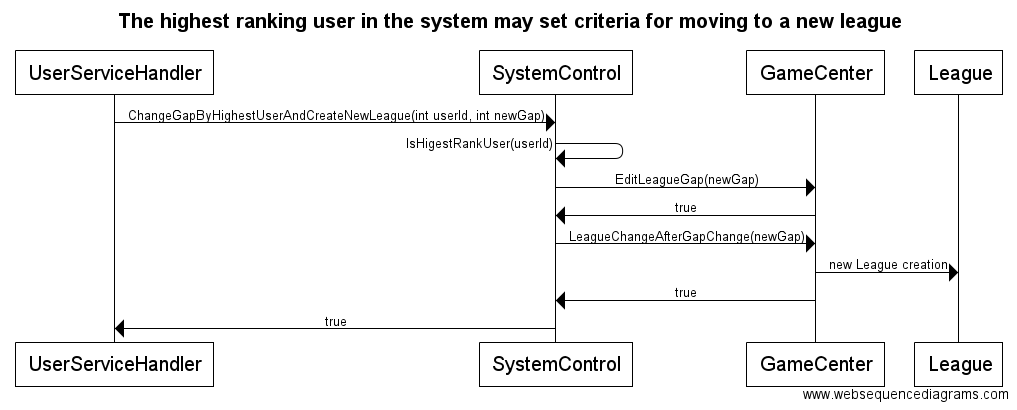
5.2.1. ביטול: המשתמש יכול לבטל את קביעת הקריטריון למעבר לליגה  
 טרם סיים לעדכנם. במקרה זה הקריטריון למעבר לליגה חדשה לא   
 נקבע.  
 5.2.2. הזנת קריטריון שגוי: במהלך שלב הזנת הקריטריון, המערכת מזהה  
 כי הקריטריון שהוזן שגוי (מבחינת Syntax או מבחינת חוקיות לוגית   
 של המשחק). במקרה זה המערכת תודיע למשתמש והקריטריון לא  
 ייקבע.

תרחיש בדיקות קבלה:

1. Use case: קביעת קריטריון למעבר לליגה חדשה.
2. Good:
   1. קביעת קריטריון מוצלחת:

המשתמש המבקש לשנות את הקריטריון הינו משתמש חבר בעל הדירוג הגבוהה ביותר במערכת והינו מחובר. הקריטריונים המוזנים על ידיו הינם חוקיים, המערכת מאשרת את שינוי הקריטריון ומשנה את הקריטריונים בליגות. המשתמשים הקיימים מחולקים מחדש בין הליגות.

1. Sad:
   1. חוסר מידע:   
      המשתמש המבקש לשנות את הקריטריון הינו משתמש חבר בעל הדירוג הגבוהה ביותר במערכת והינו מחובר, אך אינו מכניס את כל המידע הנדרש על מנת לבצע את עדכון הקריטריונים. מוצגת למשתמש הודעת שגיאה, שום מידע לא משתנה במערכת, המשתמש נשאר בתרחיש שימוש 5.1.3.
   2. משתמש לא רשאי לבצע את העדכון:  
      המשתמש אינו המשתמש בעל הדירוג הגבוהה ביותר במערכת. המערכת מציגה למשתמש הודאת שגיאה. שום מידע אינו משתנה במערכת. המשתמש נשאר בתרחיש שימוש 5.1.1.
2. Bad:
   1. מידע לא חוקי:  
      המשתמש הזין מידע אשר אינו חוקי בקביעת הקריטריון. המערכת מודיעה על כך למשתמש. שום מידע אינו משתנה במערכת. המשתמש נשאר בתרחיש שימוש 5.1.3.



**DELETE**

**The highest ranking user in the system may move users between leagues**

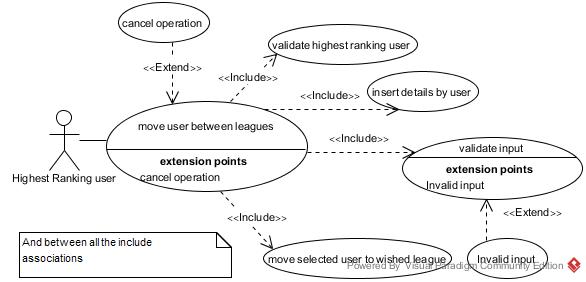
**ניתוח דרישות \ הנחות:**1) ליגת שחקן מאופיינת בטווח דירוגים (rank), פעולת העברת שחקנים בין ליגות שונות תתבצע   
 ע"י שינוי דירוגם.  
2) רק המשתמש בעל ה-Rank הגבוה ביותר רשאי להעביר שחקנים בין ליגות.

**Use Case:**

1. שם: העברת משתמשים בין ליגות.  
   1.1. תיאור: המשתמש בעל הדירוג הגבוה ביותר מעביר משתמש לליגה ספציפית.
2. משתתפים: עקרי – השחקן בעל בדירוג הגבוה ביותר, משני - מערכת המשחק, השחקן לו משנים את הליגה.
3. תנאי קדם: המשתמש המבקש לבצע את הפעולה מחובר למערכת באופן תקין וחוקי והוא המשתמש בעל הדירוג הגבוהה ביותר במערכת.
4. תנאי בתר: השחקן הנבחר עובר לליגה אחרת לבחירת השחקן בעל הדירוג הגבוה ביותר.
5. מהלך האירועים:
   1. מהלך בסיסי:  
      5.1.1. המשתמש מבקש להעביר שחקן ספציפי אחר לליגה ספציפית מסוימת.   
      5.1.2. המערכת בודקת האם המשתמש הוא המשתמש בעל הדירוג הגבוה  
       ביותר.  
      5.1.3. המערכת מבקשת מהמשתמש להזין את מזהה השחקן הרצוי ואת הליגה  
       המבוקשת להעברתו.  
      5.1.4. המשתמש מזין את הנתונים.  
      5.1.5. המערכת בודקת כי הנתונים שהוזנו חוקיים ותקינים.  
      5.1.6. המערכת מאשרת את הנתונים, ומעבירה את השחקן הנבחר לליגה  
       הרצויה.

5.2. מהלך אלטרנטיבי:

5.2.1. ביטול: המשתמש יכול לבטל את המעבר. במקרה זה הפעולה לא  
 תתבצע.  
 5.2.2. הזנת קריטריון שגוי: במהלך שלב הזנת הנתונים, המערכת מזהה  
 כי קריטריון שהוזן שגוי (מבחינת Syntax או מבחינת חוקיות לוגית   
 של המשחק). במקרה זה המערכת תודיע למשתמש והפעולה לא   
 תתבצע.



**Search/ filter active games by: player name/ pot size/ game preference**

**ניתוח דרישות \ הנחות:**

1. חיפוש לפי מאפיין משחק הוא על פי מאפיין אחד בלבד, ולא קומבינציות של אלה.

**Use Case:**

1. שם: חיפוש משחקים פעילים לפי שם שחקן\pot size\ מאפיין משחק.  
   1.1. תיאור: משתמש מכניס פילטר רצוי ומערכת המשחקים מציגה לו את כל המשחקים הפעילים המתאימים אחרי הסינון.
2. משתתפים: עיקרי - משתמש, משני - מערכת המשחקים.
3. תנאי קדם: המשתמש מחובר למערכת המשחק באופן חוקי.
4. תנאי בתר: מוצגים למשתמש כל המשחקים הפעילים המתאימים לחיפוש.
5. מהלך האירועים:
   1. מהלך בסיסי:
      1. המשתמש בוחר את הפילטר הרצוי.
      2. המשתמש מזין את ערך הפילטר.
      3. המערכת מחפשת את כל המשחקים הפעילים המתאימים לחיפוש הרצוי.
      4. המערכת מציגה לשחקן את משחקים אלה.
   2. מהלך אלטרנטיבי:

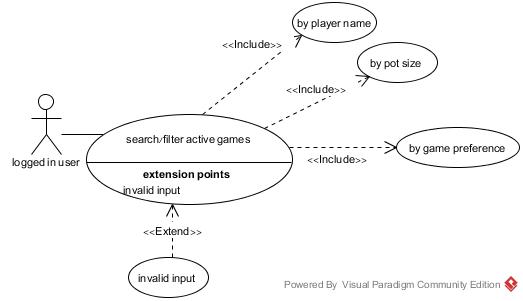
5.1.2. נתונים שגויים: המערכת מתריעה על כך למשתמש ולא מציגה לו דבר.

תרחיש בדיקות קבלה:

1. Use case: חיפוש משחקים פעילים לפי שם שחקן\pot size\ מאפיין משחק.
2. Good:
   1. חיפוש מוצלח:

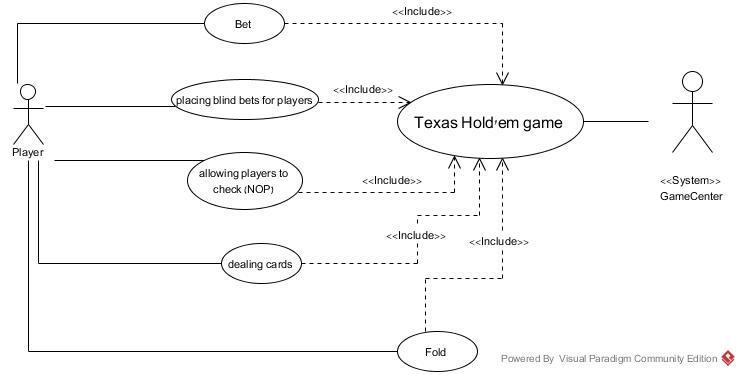
המשתמש מבקש לחפש משחקים פעילים ע"פ פילטר מרשימת הפילטרים. ומזין את ערך הפילטר בערך חוקי. המערכת מציגה למשתמש את רשימת המשחקים לאחר הסינון המבוקש.

1. Sad:
   1. חוסר מידע:   
      המשתמש מבקש לחפש משחקים פעילים ע"פ פילטר מרשימת הפילטרים, אך אינו מזין את ערך הפילטר כלל. המערכת מציגה למשתמש הודעת שגיאה. המשתמש נשאר בתרחיש שימוש 5.1.2.
2. Bad:
   1. ערך לא חוקי:  
      המשתמש מבקש לחפש משחקים פעילים ע"פ פילטר מרשימת הפילטרים, אך מזין ערך הפילטר לא חוקי. המערכת מציגה למשתמש הודעת שגיאה. המשתמש נשאר בתרחיש שימוש 5.1.2.



**Support playing a Texas Hold'em game: dealing cards, placing blind bets for players, allowing players to check (NOP), fold or bet according to the game rules, etc.**

**תמיכה בפעולות הבאות:  
בהמשך ישנו פירוק של פעולות אלה.**



**Dealing cards (Support playing a Texas Hold'em game):**

**ניתוח דרישות \ הנחות:**

1. המערכת מכירה את לוגיקת המשחק עבור כל משחק ויודעת כמה קלפים כל שחקן צריך לקבל.
2. המערכת יודעת להבדיל בין שחקן-צופה לשחקן-משחק, ויודעת שצריך לחלק קלפים רק לשחקן משחק.
3. המערכת מחלקת את הקלפים באופן רנדומלי.

**Use Case:**

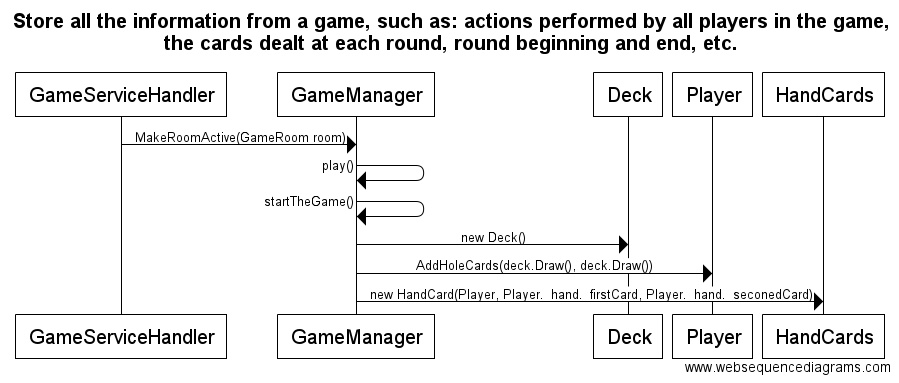
1. שם: חלוקת קלפים.  
   1.1. תיאור: מערכת המשחק מחלקת לשחקנים המשחקים במשחק ספציפי קלפים טרם תחילת המשחק.
2. משתתפים: עיקרי – מערכת המשחקים, משני - השחקנים המשחקים המשחק ספציפי, שחקן התפקיד הדילר.
3. תנאי קדם: כל שחקני המשחק מחוברים למערכת המשחק ולחדר המשחק באופן חוקי. המשחק יכול להתחיל באופן חוקי מבחינה לוגית.
4. תנאי בתר: המשחק מתחיל לאחר חלוקת הקלפים בצורה חוקית לכל השחקנים המשחקים במשחק.
5. מהלך האירועים:
   1. מהלך בסיסי:
      1. המשחק מתחיל באופן חוקי.
      2. המערכת מחלקת לכל השחקנים המשחק קלפים בצורה רנדומלית לפי חוקי המשחק – שחקן הSmall blind מקבל את הקלף הראשון ודילר את הקלף האחרון.
      3. המשחק מקבל הרשאה לתחילתו; השחקנים רשאים להתחיל.
   2. מהלך אלטרנטיבי:

5.1.2. אחד השחקנים יוצא מהמשחק בזמן חלוקת הקלפים: המקרה זה, פעולת החלוקה מתבטלת כולה, השחקנים מקבלים התראה על כך, והמשחק אינו מתחיל כלל.

**בדיקות קבלה:**

1. Use case: Dealing Cards*.*
2. Good:
   1. חלוקת קלפים תקינה:  
      ישנם מספיק שחקנים פעילים בחדר על פי הגדרות החדר, חבילת הקלפים מעורבבת, מתחלקים שני קלפים לכל שחקן כך שהשחקן שבעמדת ה-Small Blind מקבל את הקלף הראשון והשחקן שבעמדת ה-Dealer מקבל את הקלף האחרון. מספר הקלפים בחבילה קטן במספר השווה לפעמיים מספר השחקנים בחדר. מספר הקלפים של כל שחקן שווה בדיוק לשניים.
3. Sad:
   1. שחקן יוצא מהמשחק במהלך חלוקת הקלפים:  
      שחקן כלשהו יוצא מן החדר במהלך חלוקת הקלפים. המערכת מבטלת את חלוקת הקלפים, השחקן יוצא מן החדר, החבילה מעורבבת מחדש. המשחק אינו מתחיל.

תרשים רצף:



**Support playing a Texas Hold'em game: placing blind bets for players**

**ניתוח דרישות \ הנחות:**

1. המערכת יודעת לחשב את הbig blind ואת ה- small blind ויודעת את מיקומי השחקנים הללו:  
   small blind: השחקן שמשמאל לדילר, שווה לחצי מה-big blind.  
   big blind: השחקן שמשמאל לשחקן ה-small blind, שווה ל-bet המינימלי.
2. המערכת יודעת לזהות את מיקום הדילר.
3. כשמגיעים למצב בו יש רק שני שחקנים, הדילר הוא ה-small blind, והשחקן השני הוא ה-big blind.

**Use Case:**

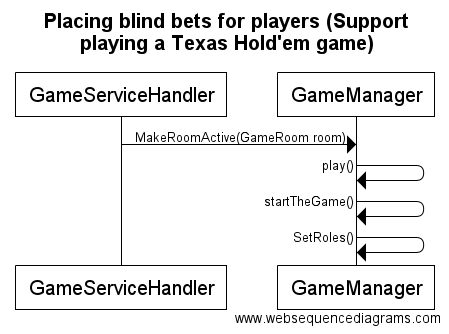
1. שם: למקם ולשים את ה-blind bets לשחקנים.  
   1.1. תיאור: המערכת ממקמת את הblind bets ומיישמת את ההימור לשחקנים המתאימים.
2. משתתפים: עיקרי: שחקני המשחק, מערכת המשחק. משני – המשחק.
3. תנאי קדם: השחקנים המשחקים מחוברים למערכת באופן תקין וחוקי. המשחק מתנהל באופן חוקי ותקין. השחקן המתאים הוגדר להיות הדילר.
4. תנאי בתר: השמת ה-blind bets לשחקנים המתאימים.
5. מהלך האירועים:
   1. מהלך בסיסי:  
      5.1.1. מערכת המשחק תבקש בקשה מהמשחק הפעיל למקם את ה-blind bets.   
      5.1.2. מערכת המשחק תבחר את השחקן שמשמאל לדילר להיות ה-small blind.  
      5.1.3. מערכת המשחק מחשבת את סכום ה-Small blind.  
      5.14. מערכת המשחק מתריעה על כך לשחקן הSmall blind, ולוקחת ממנו הסכום המתאים.  
      5.1.5. מערכת המשחק תבחר את השחקן שמשמאל לשחקן הsmall blind להיות שחקן ה-big blind.  
      5.1.6. מערכת המשחק מחשבת את סכום ה-Big blind.  
      5.1.7. מערכת המשחק מתריעה על כך לשחקן הBig blind, ולוקחת ממנו הסכום המתאים.  
      5.1.8. המערכת מתריעה למשחק כי ה-blind bets יושמו והונחו לשחקנים המתאימים.

**בדיקות קבלה:**

1. Use case: *Placing blind bets.*

2. Good:

* 1. . ביצוע Blind bets:  
     המשחק התחיל, חולקו קלפים לשחקנים, לשחקן בעמדת ה-Small Blind הורדו כמות ג'יטונים השווה לסכום הSmall Blind, שחקן זה הינו השחקן משמאלו של ה-Dealer. לשחקן בעמדת ה-Big Blind הורדו כמות ג'יטונים השווה לפעמיים סכום הSmall Blind, שחקן זה הינו השחקן משמאלו של השחקן שבעמדת ה-Small Blind. התור הנוכחי הוא של השחקן שמשמאלו של השחקן שבעמדת ה-Big Blind. המערכת מודיעה לשחקנים כי ה-Blind Bets הונחו.

****

**Fold according to the game rules (Support playing a Texas Hold'em game):**

**ניתוח דרישות \ הנחות:**

1. המערכת מכירה את לוגיקת המשחק עבור פעולת ה-Fold.

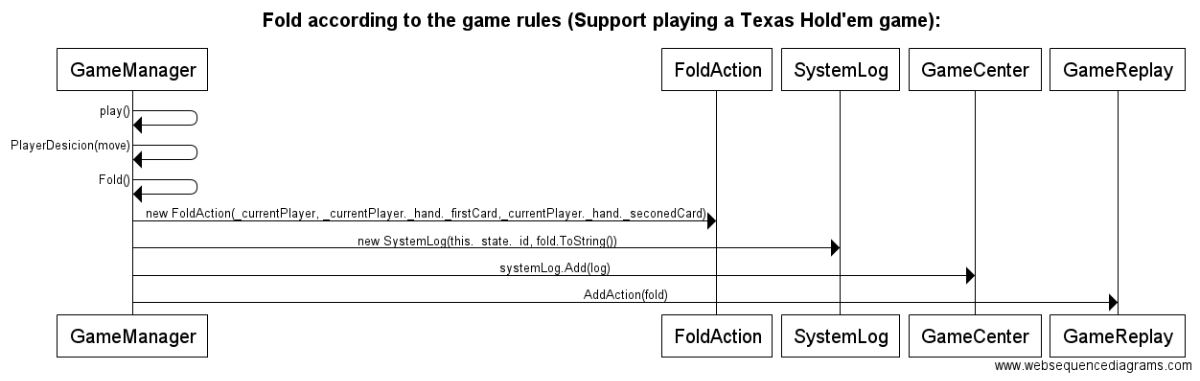
**Use Case:**

1. שם: ביצוע פעולת Fold ע"י משתמש בהתאם לכללי המשחק.  
   1.1. תיאור: שחקן מבקש לבצע פעולת FOLD, המערכת מוודאת שניתן עפ"י חוקי המשחק, ומאפשרת ומוציאה לפועל פעולה זאת.
2. משתתפים: עיקרי :שחקן, מערכת המשחק, משני: משחק.
3. תנאי קדם: השחקן במשחק מחובר למערכת המשחק ולחדר המשחק באופן חוקי. המשחק פועל באופן תקין וחוקי מבחינה לוגית.
4. תנאי בתר: השחקן מבצע את פעולת הFOLD בהתאם לכללי המשחק.
5. מהלך האירועים:
   1. מהלך בסיסי:  
      5.1.1. השחקן מבקש מהמשחק לבצע פעולת FOLD.  
      5.1.2. המשחק מבקש ממערכת המשחק לבדוק האם פעולה זאת חוקית.  
      5.1.3. מערכת המשחק מאשרת כי הפעולה המבוקשת חוקית.  
      5.1.4. המשחק מבצע את פעולת ה-FOLD עבור השחקן ומתריעה לו על כך.
   2. מהלך אלטרנטיבי:

5.1.2. פעולת FOLD אינה חוקית: השחקן מנסה לבצע FOLD כנגד לוגיקת המשחק.   
 במצב זה הפעולה אינה מתבצעת והמשתמש מקבל התרעה על כך.

**בדיקות *קבלה:***

1. Use case: *Fold according to the game rules.*
2. Good:
   1. ביצוע פעולת Fold תקינה:  
      השחקן המבקש לבצע את הפעולה הינו השחקן שתורו הוא הנוכחי והוא רשאי לבצעה על פי חוקי המשחק. המערכת מבצעת פעולת Fold, המידע בחדר המשחק משתנה בהתאם, התור עובר לשחקן הבא באם יש.
3. Sad:
   1. השחקן אינו רשאי לבצע Fold:  
      השחקן המבקש לבצע את הפעולה הינו השחקן שתורו הוא הנוכחי, אך הוא אינו רשאי לבצע Fold על פי חוקי המשחק. המערכת מתריעה לו כי אינו רשאי לבצע פעולה זו, שום מידע אינו משתנה בחדר המשחק, התור הנוכחי הוא עדין של השחקן.
4. Bad:
   1. לא תורו של השחקן:  
      השחקן המבקש לבצע פעולה זו אינו השחקן אשר תורו הוא הנוכחי. המערכת מודיעה לו כי התור הנוכחי אינו שלו. שום מידע אינו משתנה בחדר המשחק.



**Allowing players to check (NOP) (Support playing a Texas Hold'em game)**

**ניתוח דרישות\הנחות:**

1. המערכת מכירה את לוגיקת המשחק: פעולת ה-check ע"י השחקן יכולה להתבצע אך ורק אם כל השחקנים השוו לbig blind.

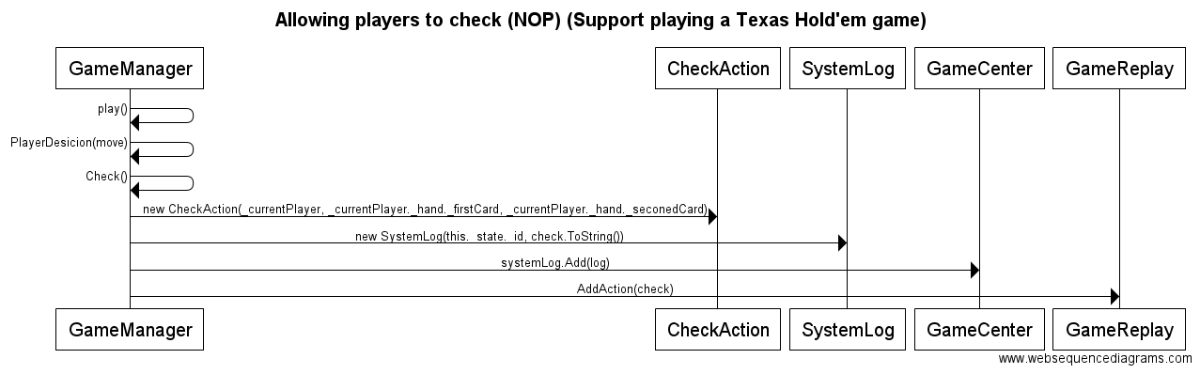
**Use Case:**

1. שם: אפשרת ביצוע פעולת ה-check לשחקן.  
   1.1. תיאור: בהינתן שחקן המבקש לבצע את פעולת ה-check, במידה וניתן לבצע אותה חוקית מבחינת לוגיקת המשחק, המערכת מאפשרת לו זאת.
2. משתתפים: עיקרי - שחקן, מערכת המשחק, משני - משחק.
3. תנאי קדם: המשתמש המבקש לבצע את הפעולה מחובר למערכת באופן תקין וחוקי.
4. תנאי בתר: המשתמש ביצוע פעולת check בצורה תקינה וחוקית.
5. מהלך האירועים:  
   5.1. מהלך בסיסי:
   * 1. השחקן מבקש מהמשחק לבצע פעולת Check.
     2. המשחק מבקש ממערכת המשחק לבדוק האם פעולה זאת חוקית.
     3. מערכת המשחק מאשרת כי הפעולה המבוקשת חוקית.
     4. המשחק מבצע את פעולת ה-check עבור השחקן ומתריעה לו על כך.
   1. מהלך אלטרנטיבי:
      1. פעולת הcheck אינה חוקית:  
         המקרה זה, המערכת מודיעה לשחקן כי פעולה זו אינה חוקית, המערכת אינה מבצעת את הפעולה הנ"ל ומתריעה לשחקן על כך.

***ב*דיקות קבלה:**

1. Use case: *Allowing players to check.*
2. Good:
   1. ביצוע פעולת Check תקינה:  
      השחקן המבקש לבצע את הפעולה הינו השחקן שתורו הוא הנוכחי והוא רשאי לבצעה על פי חוקי המשחק. המערכת מבצעת פעולת Check, המידע בחדר המשחק משתנה בהתאם, התור עובר לשחקן הבא באם יש.
3. Sad:
   1. השחקן אינו רשאי לבצע Check:  
      השחקן המבקש לבצע את הפעולה הינו השחקן שתורו הוא הנוכחי, אך הוא אינו רשאי לבצע Check על פי חוקי המשחק. המערכת מתריעה לו כי אינו רשאי לבצע פעולה זו, שום מידע אינו משתנה בחדר המשחק, התור הנוכחי הוא עדין של השחקן.
4. Bad:
   1. לא תורו של השחקן:  
      השחקן המבקש לבצע פעולה זו אינו השחקן אשר תורו הוא הנוכחי. המערכת מודיעה לו כי התור הנוכחי אינו שלו. שום מידע אינו משתנה בחדר המשחק.

תרשים רצף:



**Call according to the game rules (Support playing a Texas Hold'em game):**

**ניתוח דרישות \ הנחות:**

1. המערכת מכירה את לוגיקת המשחק עבור פעולת ה-call – השוואת ההימור של השחקן הקודם בסיבוב.

**Use Case:**

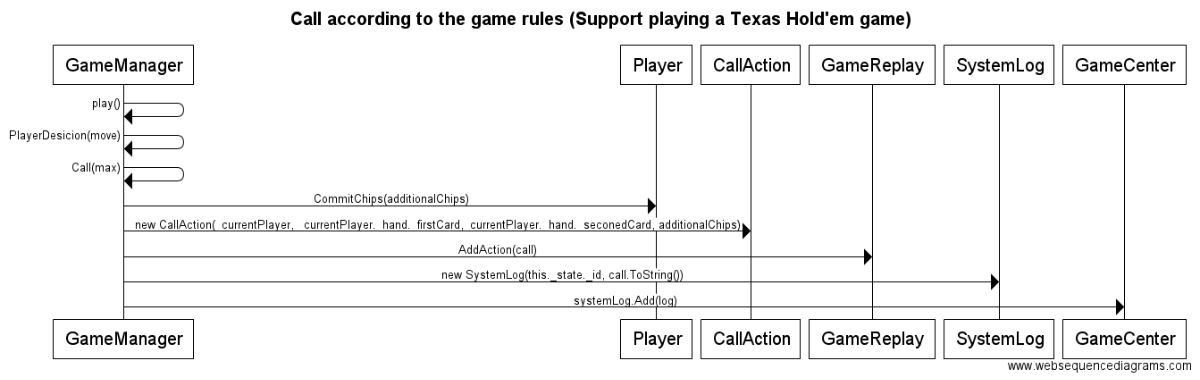
1. שם: ביצוע פעולת Call ע"י משתמש בהתאם לכללי המשחק.  
   1.1. תיאור: שחקן מבקש לבצע פעולת call, המערכת מוודאת שניתן עפ"י חוקי המשחק, ומאפשרת ומוציאה לפועל פעולה זאת.
2. משתתפים: עיקרי -שחקן, מערכת המשחק, משני -משחק.
3. תנאי קדם: השחקן במשחק מחובר למערכת המשחק ולחדר המשחק באופן חוקי. המשחק פועל באופן תקין וחוקי מבחינה לוגית.
4. תנאי בתר: השחקן מבצע את פעולת הcall בהתאם לכללי המשחק.
5. מהלך האירועים:
   1. מהלך בסיסי:  
      5.1.1. השחקן מבקש מהמשחק לבצע פעולת call.  
      5.1.2. המשחק מבקש ממערכת המשחק לבדוק האם פעולה זאת חוקית.  
      5.1.3. מערכת המשחק בודקת האם השחקן יכול לבצע פעולה זאת (בהתאם לנתוניו(  
      5.1.3. מערכת המשחק מאשרת כי הפעולה המבוקשת חוקית.  
      5.1.4. המשחק מבצע את פעולת ה-Call עבור השחקן; מורידה לו את הסכום המתאים.  
      5.1.6. המשחק מתריע לשחקן כי פעולתו התבצעה בהצלחה.
   2. מהלך אלטרנטיבי:

5.1.2. פעולת Call אינה חוקית: השחקן מנסה לבצע Call כנגד לוגיקת המשחק.   
 במצב זה הפעולה אינה מתבצעת והמשתמש מקבל התרעה על כך.

***ב*דיקות קבלה:**

1. Use case: *Allowing players to call.*
2. Good:
   1. ביצוע פעולת Call תקינה:  
      השחקן המבקש לבצע את הפעולה הינו השחקן שתורו הוא הנוכחי והוא רשאי לבצעה על פי חוקי המשחק. המערכת מבצעת פעולת Call, המידע בחדר המשחק משתנה בהתאם, התור עובר לשחקן הבא באם יש.
3. Sad:
   1. השחקן אינו רשאי לבצע Call:  
      השחקן המבקש לבצע את הפעולה הינו השחקן שתורו הוא הנוכחי, אך הוא אינו רשאי לבצע Cal על פי חוקי המשחק. המערכת מתריעה לו כי אינו רשאי לבצע פעולה זו, שום מידע אינו משתנה בחדר המשחק, התור הנוכחי הוא עדין של השחקן.
4. Bad:
   1. לא תורו של השחקן:  
      השחקן המבקש לבצע פעולה זו אינו השחקן אשר תורו הוא הנוכחי. המערכת מודיעה לו כי התור הנוכחי אינו שלו. שום מידע אינו משתנה בחדר המשחק.

תרשים רצף:



**Raise according to the game rules (Support playing a Texas Hold'em game):**

**ניתוח דרישות \ הנחות:**

1. המערכת מכירה את לוגיקת המשחק עבור פעולת ה- Raise.

**Use Case:**

1. שם: ביצוע פעולת Raise ע"י משתמש בהתאם לכללי המשחק.  
   1.1. תיאור: שחקן מבקש לבצע פעולת Raise, המערכת מוודאת שניתן עפ"י חוקי המשחק, ומאפשרת ומוציאה לפועל פעולה זאת.
2. משתתפים: עיקרי - שחקן, מערכת המשחק, משני - משחק.
3. תנאי קדם: השחקן במשחק מחובר למערכת המשחק ולחדר המשחק באופן חוקי. המשחק פועל באופן תקין וחוקי מבחינה לוגית.
4. תנאי בתר: השחקן מבצע את פעולת ה- Raise בהתאם לכללי המשחק.
5. מהלך האירועים:
   1. מהלך בסיסי:  
      5.1.1. השחקן מכניס את סכום ההעלאה הרצוי.  
      5.1.2. השחקן מבקש מהמשחק לבצע פעולת Raise.  
      5.1.3. המשחק מבקש ממערכת המשחק לבדוק האם פעולה זאת חוקית.  
      5.1.4. מערכת המשחק בודקת האם השחקן יכול לבצע פעולה זאת (בהתאם לנתוניו(  
      5.1.5. מערכת המשחק מאשרת כי הפעולה המבוקשת חוקית.  
      5.1.6. המשחק מבצע את פעולת ה- Raise עבור השחקן; מורידה לו את הסכום המתאים.  
      5.1.7. המשחק מתריע לשחקן כי פעולתו התבצעה בהצלחה.
   2. מהלך אלטרנטיבי:

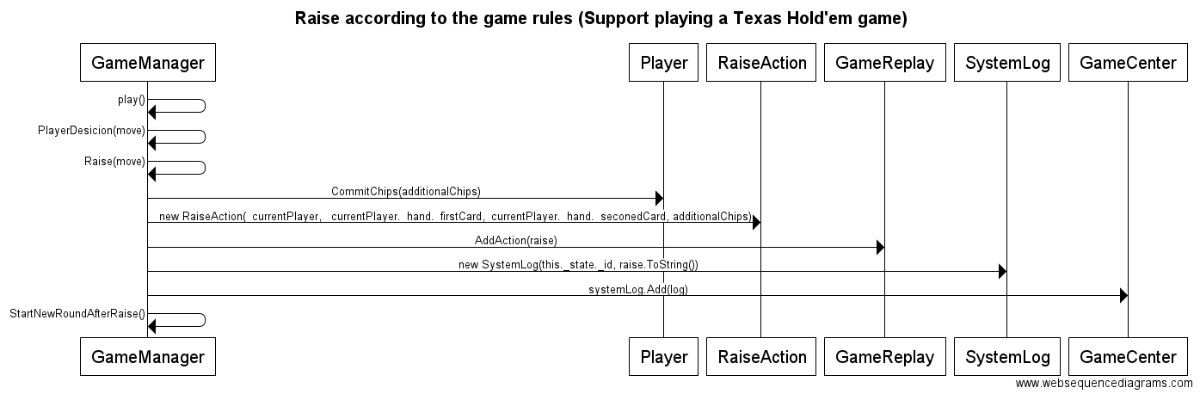
5.1.2. פעולת Raise אינה חוקית: השחקן מנסה לבצע Raise כנגד לוגיקת המשחק.   
 במצב זה הפעולה אינה מתבצעת והמשתמש מקבל התרעה על כך.

5.1.3. הנתונים שהוזנו אינם חוקיים: השחקן מזין נתונים שאינם חוקיים כאשר מבקש   
 לבצע פעולת Raise.   
 במצב זה הפעולה אינה מתבצעת והמשתמש מקבל התרעה על כך.

***ב*דיקות קבלה:**

1. Use case: *Allowing players to Raise.*
2. Good:
   1. ביצוע פעולת Raise תקינה:  
      השחקן המבקש לבצע את הפעולה הינו השחקן שתורו הוא הנוכחי והוא רשאי לבצעה על פי חוקי המשחק. המערכת מבצעת פעולת Raise, המידע בחדר המשחק משתנה בהתאם, התור עובר לשחקן הבא באם יש.
3. Sad:
   1. השחקן אינו רשאי לבצע Raise:  
      השחקן המבקש לבצע את הפעולה הינו השחקן שתורו הוא הנוכחי, אך הוא אינו רשאי לבצע Raise על פי חוקי המשחק. המערכת מתריעה לו כי אינו רשאי לבצע פעולה זו, שום מידע אינו משתנה בחדר המשחק, התור הנוכחי הוא עדין של השחקן.
   2. הסכום אינו חוקי:  
      השחקן המבקש לבצע את הפעולה הינו השחקן שתורו הוא הנוכחי אך הסכום עליו הוא מעוניין להמר אינו חוקי על י חוקי המשחק. . המערכת מתריעה לו כי אינו רשאי לבצע פעולה זו, שום מידע אינו משתנה בחדר המשחק, התור הנוכחי הוא עדין של השחקן.
4. Bad:
   1. לא תורו של השחקן:  
      השחקן המבקש לבצע פעולה זו אינו השחקן אשר תורו הוא הנוכחי. המערכת מודיעה לו כי התור הנוכחי אינו שלו. שום מידע אינו משתנה בחדר המשחק.

תרשים רצף:



**Class diagram**

**Component diagram**

