**תרגיל 1 בקורס ג'אווה ואינטרנט**

**מגישים:**

**דקל צעירי 311319057– dekeltsairi@gmail.com**

**נוי וולפסון 315373282 – noy@wolfson.co.il**

**הפעלת התוכנית**

**תחילת האפליקציה** –

1. בתחילת האפליקציה יוצג למשתמש חלון בהיר ונהיר, המתאר את מצב המערכת כולה בכל רגע נתון, ובפרט ה branch הנוכחי, הקבצים שלו, עץ הסטוריית הקומיטים וכו'.

המע' מציגה ברגע נתון את מצבה המלא של ה repository האקטיבי. כמו כן את השם והמיקום של ה repository הפעיל בכל רגע נתון.

כאמור הצגת המידע כוללת את כל הקומיטים, branch'ים, יכולת לבדוק תוכן של קומיט מסוים וכן הלאה.

המערכת מספקת ניווט בהיר, נהיר ונוח, על יידי כפתורים ותפריטים הקשר, לביצוע הפונציונאליות הבאה:

1. **עדכון שם משתמש**
2. **קריאת קובץ repository מ-XML**
3. **אתחול Repository**
4. **החלפת repository**
5. **הצגת תוכן של קומיט**
6. **הצגת כל הקבצים של ה-commit הנוכחי בליווי מידע היסטורי**
7. **הצגת מצב ה-WC**
8. **ביצוע commit**
9. **הצגת כל ה-Branch-ים במערכת**
10. **יצירת Branch חדש**
11. **מחיקת Branch קיים**
12. **בחירת Head Branch**
13. **אתחול מיקום של headBranch**
14. **עץ קומיטים**
15. **ביצוע merge**
16. **יכולות של collaboration**

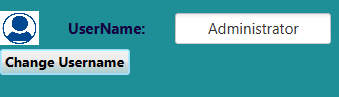
**\*את סעיפים א', ב', ג', ד', וסגירת המערכת ניתן לבצע גם דרך תפריט ההקשר בצד שמאל העליון של האפליקציה תחת File.**

**\*את כל יכולות ה- collaboration ניתן לבצע גם דרך תפריט ההקשר בצד שמאל העליון של האפליקציה תחת Collaboration.**

**פירוט כל אחת מאופציות הניתנות לבחירה באפליקציה**

1. ***עדכון שם משתמש:***

בקצה השמאלי העליון של האפליקציה, ניתן לראות את החלון הבא:



שם המשתמש הדיפפולטי הוא Administarator, לחיצה על הכפתור Change Username תאפשר למשתמש לשנות את שם המשתמש בחלון נפרד שיפתח

1. ***טעינת Repository מקובץ XML***

בצד שמאל של האפליקציה מופיעה רשימה של כפתורים, וביניהם הכפתור

לחיצה עליו תפתח למשתמש FileChooser שבו יוכל המשתמש לגשת אל מיקום קובץ ה- XML ממנו ירצה לטעון Repository

תהליך טעינת ה-Xml קורה ב-Task תחת thread נפרד, ומציג Progress bar שמראה את מצב הטעינה בכל רגע נתון.

במידה וקובץ ה- XML נמצא לא תקין, תוצג הודעה מתאימה עם הסיבה בגינה הקובץ אינו תקין. במידה ויש כמה סיבות בגינן הקובץ אינו תקין, תוצג התקלה האחרונה שהתגלתה.

במידה וקובץ ה- XML נמצא תקין:

נבדוק אם יש כבר Repository ב-location המוגדר בקובץ (האם יש באותו location ספריית .magit).

* במידה ויש שם כבר Repository אחר, נשאל את המשתמש איך ברצונו להמשיך:

1. לדרוס את ה-Repository הקיים ולייצר במקומו אחד חדש מקובץ ה-XML שהוזן.
2. להישאר עם ה- Repository הקיים ולבטל את טעינת ה- XML שהוזן.

* במידה ואין שם Repository וגם לא תוכן אחר שאינו Repository- נייצר את ה-Repository המבוקש.

בכל מקרה, בסיום תוצג למשתמש הודעה מתאימה בגין הפעולה שבוצעה.

כמו כן את הפעולה הנ"ל ניתן לעשות מתוך תפריטי ההקשר בקצה השמאלי העליון של האפליקציה

1. ***אתחול Repository:***

בצד שמאל של האפליקציה מופיעה רשימה של כפתורים, וביניהם הכפתור

לאחר לחיצה על כפתור זה, יפתח חלון חדש בו המשתמש יתבקש להזין Full Path אל המקום בו ירצה לאתחל Repository חדש.

אם המשתמש יזיו קלט לא תקין, דוגמת path קיים, הוא יקבל הודעת שגיאה מתאימה ותינתן לו הבחירה אם לנסות שוב או לחזור לתפריט.

כמו כן המשתמש יתבקש להזין שם ל-Repository החדש שיאותחל.

צליחת הפעולה תוביל להודעה מתאימה.

1. ***החלפת Repository:***

בצד שמאל של האפליקציה מופיעה רשימה של כפתורים, וביניהם הכפתור

לאחר לחיצה על כפתור זה, ייפתח חלון חדש בו המשתמש יתבקש להזין Full Path אל מיקום ה-Repository הקיים. אם המשתמש הזין Path לא קיים, או שמא Path קיים אך שאינו מהווה Repository, המשתמש יקבל הודעה מתאימה.

צליחת הפעולה תוביל להודעה מתאימה, ובסיום הפעולה האפליקציה תנהל כעת את הRepository- הנ"ל.

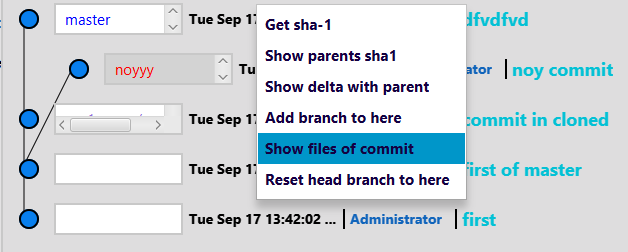
1. ***הצגת תוכן של Commit***

כאשר ישנו Repository חיי קיים ובועט, יופיע במרכז המסך עץ הקומיטים שלו.

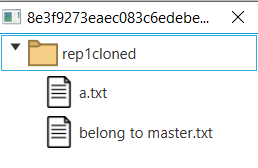
ניתן לראות בבהירות את סדר הקומיטים, ומיקומים בעץ ברור לעין.

אם המשתמש ירצה להציג את תוכנו של קומיט מסויים, כל מה שהוא צריך לעשות זה להעמד עם העכבר על הקומיט בעץ הקומיטים (אבל לא על רשימת ה-Branch-ים שלידו, כי לה יש פונקציונאליות אחרת), ללחוץ על הכפתור הימני לפתיחת תפריט הקשר, וללחוץ Show Files of Commit

תמונה להמחשת הפעולה:



מיד לאחר שהמשתמש ילחץ על Show files of commit, תופיע לו רשימה של כל הקבצים והספריות השייכים לקומיט:



לחיצה על אחד מהקבצים המסומנים כקבצי טקסט, תפתח חלון בו בו יראה המשתמש את תוכנו של הקובץ כפי שהיה בעת ביצוע ה-Commit הנ"ל.

1. ***הצגת מצב ה-WC:***

בצד שמאל של האפליקציה מופיעה רשימה של כפתורים, וביניהם הכפתור

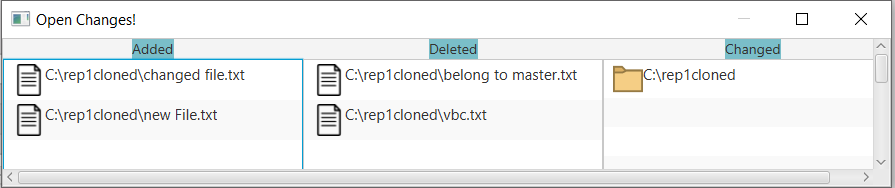
אם המשתמש ייבחר באופציה זו לאחר שטען Repository אך לפני שקיים Commit כלשהו ב-Repository שטען, יקבל המשתמש הודעה מתאימה.

אם קיים Repository טעון ואם קיים לו Commit נוכחי, אבל אם ה-WC ריק, תוצג הודעה מתאימה, כי המערכת קבצים ריקה.

אם קיים Repository טעון ואם קיים לו Commit נוכחי, וגם יש קבצים ב-WC, יוצגו כל השינויים הפתוחה ב-WC, (במידה ויש כאלה) מקוטרגים לפי:

* קבצים שנמחקו
* קבצים ששונו
* קבצים שנוספו

לדוגמא, חלון אפשרי שייפתח ייראה כך:



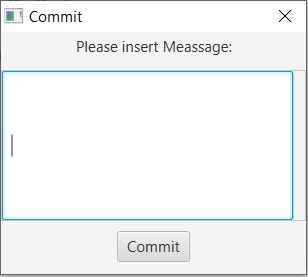
1. ***ביצוע קומיט:***

בצד שמאל של האפליקציה מופיעה רשימה של כפתורים, וביניהם הכפתור

אם המשתמש ייבחר באופציה זו לאחר שטען Repository אך לפני שקיימים קבצים כלשהם ב-WC, יקבל המשתמש הודעה מתאימה, "לא ניתן לבצע Commit למערכת קבצים ריקה".

אם קיים Repository וגם יש קבצים ב-WC, אבל ה-Commit הנוכחי מייצג בדיוק אותה מערכת קבצים (המצב נקי), יקבל המשתמש הודעה מתאימה גם כן.

אם קיים Repository וגם יש קבצים ב-WC, וגם ה-Commit הנוכחי אינו מייצג את אותה מערכת קבצים (שינויים פתוחים), או שמא לא קיים עדיין Comit נוכחי, אזי יפתח למשתמש חלון ובו יבקש להכניס את הודעת הקומיט.



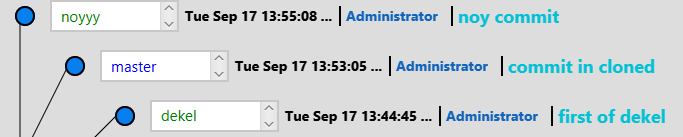
לאחר שיזין המשתמש את הודעתו, וילחץ Commit, תהליך ביצוע ה-Commit יחל והמשתמש יקבל הודעה מתאימה בדבר הצלחת ה-Commit.

1. ***הצגת כל ה-Branch-ים במערכת:***

כאמור, בכל רגע נתון תופיע בצד ימין של האפליקציה רשימת ה-Branch-ים לרבות חיווי מתאים ל-Head Branch וחיווים נוספים לרבות RTB ו-Remote Branch במקרה שמדובר ב-LL.



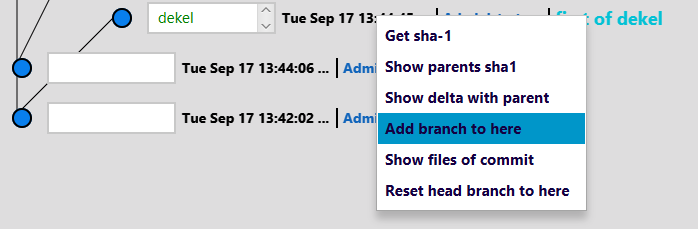
שיוכם של ה-Branch ל-Commit-ים עליהם הם מצביעים מופיע בבירור בעץ הקומיטים, עליו נרחיב בהמשך:



1. ***יצירת Branch חדש על Commit נבחר:***

בניגוד לתרגיל הקודם, בתרגיל זה יש למשתמש היכולת להוסיף Branch כלשהו על כל Commit שיחפוץ, ובלבד ששם ה-Branch שהמתמש יכניס הוא שם שאינו קיים במערכת:

הפעולה מתבצעת באמצעות עמידה עם העכבר על Commit כלשהו בעץ ה-Commit-ים, לחיצה על כפתור הימני לפתיחת תפריט הקשר, ולחיצה על Add Branch To Here :



לאחר הלחיצה, המשתמש יתבקש להזין שם של Branch .

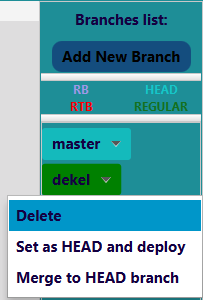
כמו כן אם שמו של ה-Branch קיים במערכת, יקבל המשתמש הודעה מתאימה.

1. ***מחיקת Branch קיים:***

כאמור, בצד ימין של האפליקציה מופיעה רשימת ה-Branch-ים.

יוכל המשתמש להעמד עם העכבר על כל Branch ברשימה, לחיצה על Branch זה תפתח תפריט הקשר שממנו יוכל לבצע את מחיקת ה-Branch עליו נעמד.

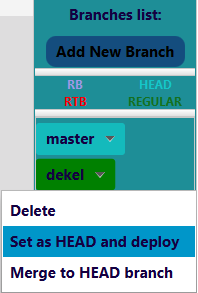
כמו כן, אם יפתח המשתמש תפריט הקשר מתוך הHead Branch, הכפתור Delete יהיה חסום כך שלא ניתן לבצע מחיקה ל -Head Bracnh:



1. ***בחירת Head Branch:***

בצד ימין של האפליקציה כאמור מופיעה רשימת ה-Branch-ים,

המשתמש יוכל לפתוח את תפריט ההקשר מתוך Branch כלשהו ולקבוע אותו כ-Head Branch :

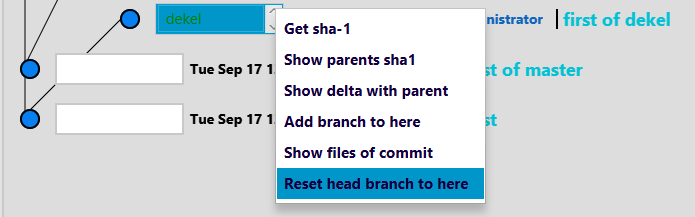


1. ***אתחול מיקום של Branch:***

המשתמש יוכל בכל רגע נתון שבו קיימים Branch-ים במערכת,להעמד על אחד ה-Comitt-ים בעץ Commit-ים, לפתוח את תפריט ההקשר וללחוץ Reset Head Branch to here.

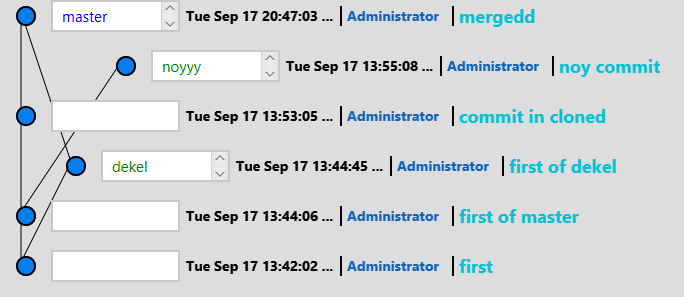
במידה וקיימים שינויים פתוחים במערכת, תוצג למשתמש הודעה מתאימה הנותנת לו את האפשרויות הבאות:

* להמשיך ולאבד את השינויים הפתוחים
* לבטל את הפעולה



1. ***עץ הקומיטים:***

לפי דרישת התרגיל, קיים באפליקציה עץ קומיטים המופיע בכל רגע נתון.



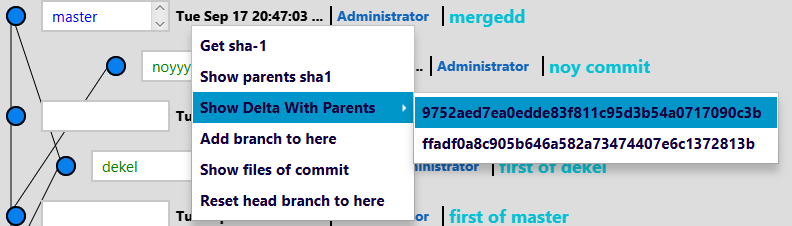
המשתמש יוכל לבחור כל קומיט בעץ, ובאמצעות תפריט ההקשר לקבל מידע הנ"ל לגביו

* SHA-1
* הודעת קומיט
* מי יצר אותו
* מתי נוצר
* SHA-1 של קומיט(ים) קודמ(ים) (הורים ישירים בלבד)
* פרטים על המידע שהשתנה בקומיט זה (דלתא מול קומיט קודם)

יש להציג זאת כרשימה של שינויים, כפי שמציגים את פירוט ה WC יחסית לקומיט הנוכחי:

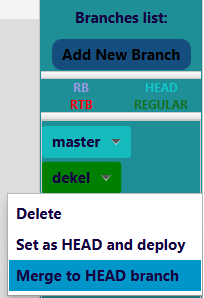
* שם הקובץ שהשתנה
* האם עודכן/נוצר/נמחק
* אם מדובר בקומיט שהוא תוצאה של merge, הרי שאז יש לו 2 אבות,

במצב זה תינתן למשתמש האפשרות לבחור איזו דלתא הוא רוצה לחשב, מול איזה אב, בדמות תפריט הקשר נוסף שיאפשר לו לבחור את קומיט האבא שמולו הוא רוצה לראות את השינויים.



1. ***ביצוע Merge:***

למשתמש יש האפשרות להעמד על אחד מה-Branch-ים ברשימת ה-Branch-ים שאינו ה-Head ולבצע את הפעולה Merge to Head Branch מתוך תפריט ההקשר.

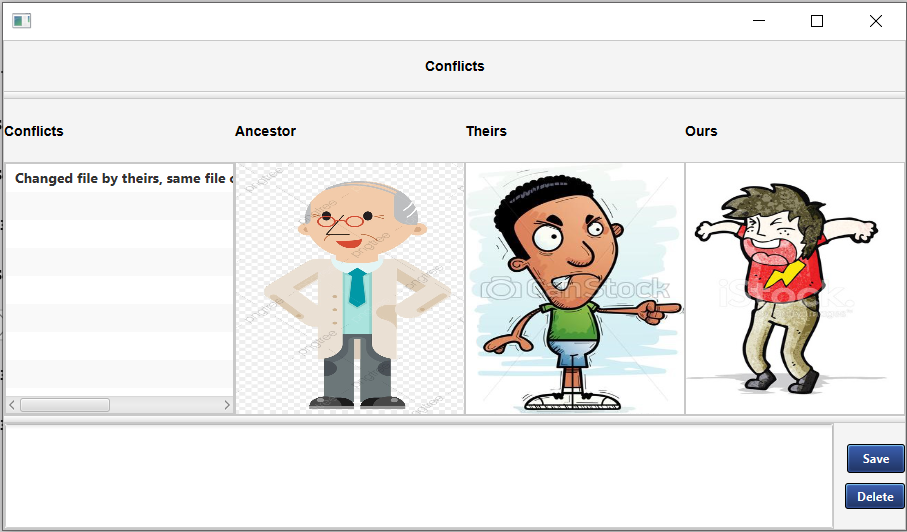


אם יש שינויים פתוחים, לאחר לחיצה על הכפתור הנ"ל המשתמש יקבל הודעה מתאימה.

לא ניתן לבצע Merge כאשר יש שינויים פתוחים.

במידה ולא, תהליך ה-Merge יחל, וייתכנו כמה תרחישים:

1. אם מדובר ב-Fast Merge המשתמש יקבל הודעה מתאימה
2. אם מדובר ב-Merge רגיל אך חסר קונפליקטים, המתמש יתבקש להכניס הודעה לחלון הCommit המוכר מביצוע Commit ולאחר פעולה זאת תהליך ה-Merge ייסתיים
3. במידה וקיימים קונפליקטים, למשתמש יוצג חלון שבו יוכל להתחיל לפתור את הקונפליקטים כפי שמתואר באיפון התרגיל:



המשתמש יוכל לנווט ברשימת כל הקונפליקטים, לראות למעלה הודעה המתארת את הקונפליקט, לרשום את התוכן אותו החדש בחלון התחתון, איתו הוא רוצה לפתור את ה-Conflict, לבצע save ולעבור ולפתור את הקונפליקט הבא.

לחילופין, יכול המשתמש לפתור קונפליקט כלשהו על ידי מחיקתו.

לאחר שפתר את כל הקונפליקטים, המשתמש יעבור למסך ה-Commit המוכר, בו יזין הודעת קומיט, ילחץ Commit ויסיים את תהליך ה-Merge

1. ***Collaboration***

בתרגיל זה אנו תומכים בעבודה ובשיתוף מידע עם repositories אחרים, "מרוחקים".

נפרט את 4 הפעולות העיקריות של יכולות זו, הנקראת Collaboration:

1. Clone

היכולת לייצר repository חדש על ידי שכפול repository קיים.

בצד שמאל של האפליקציה מופיעה רשימה של כפתורים, וביניהם הכפתור

לאחר לחיצה על כפתור זה, למשתמש ייפתח חלון חדש בו יכניס נתיב מלא לספרייה קיימת של M.A.Git (ה RR) כמו גם נתיב מלא לספריית היעד אליה הוא ירצה לשכפל את ה repository (ה LR). בנוסף יכניס המשתמש את שם ה repository החדש שיווצר.

השכפול הוא מלא, וכולל את כל המידע שיש ב repository המקורי: כל הקבצים והספריות לדורותיהם, כל הקומיטים וכל ה branch'ים.

עבור כל branch ב RR יווצר RB ב repository המשוכפל, היסומן בצבע אחר.

כמו כן, אם המשתמש יכניס איזשהו נתיב שלא עונה על הדרישה, יקבל המשתמש הודעת שגיאה מתאימה.

עבור ה HEAD ב RR יווצר RTB אוטומטית מתאים ב LR והוא כמובן יהיה ה HEAD ב LR. שמו של ה RTB, כאמור, יהיה בדיוק כמו שמו של ה branch ב RR...

1. Fetch

בצד שמאל של האפליקציה מופיעה רשימה של כפתורים, וביניהם הכפתור

כפתור זה יהיה לחיץ אך ורק עם ה-Repository הנוכחי הטעון במערכת הוא LR

לאחר לחיצה על כפתור זה, המערכת מייבאת את כל המידע החדש שהצטבר ב RR אל ה LR.

לאחר פקודה זו ה RR מסונכרן מול ה LR

המידע החדש כולל branch'ים חדשים שנוצרו, עדכונים למיקומים של branch'ים קיימים, קומיטים, ספריות וקבצים חדשים שהתווספו.

לאחר פקודה זו כל ה RBs יתעדכנו למיקומים החדשים שלהם כפי שהם מוצגים כרגע ב RR.

ה RTB נותרים במקומם !

1. Pull

בצד שמאל של האפליקציה מופיעה רשימה של כפתורים, וביניהם הכפתור

כפתור זה יהיה לחיץ אך ורק עם ה-Repository הנוכחי הטעון במערכת הוא LR

לאחר להחיצה על הכפתור, נייבא מה RR **רק** את הקבצים, ספריות, קומיטים של ה branch שהוא ה-Head Branch הנוכחי. לאחר מכן נעדכן את ה RB, כמו גם את RTB להצביע על המיקומים הרלבנטים שלהם.

מטבעה של פעולה זו, פעולת ה-Pull תכנס לתוקף אך ורק אם:

* ה head branch הוא כמובן RTB אחרי RB אחר.
* המערכת נקייה משינויים.
* אין עוד שינויים ב LR על ה RTB שטרם עברו Push (פקודה הבאה)

1. Push

בצד שמאל של האפליקציה מופיעה רשימה של כפתורים, וביניהם הכפתור

כפתור זה לחיץ אך ורק עם ה-Repository הטעון במערכת הינו LR

פקודה זו דוחפת מידע חדש של ה-Head Branch (קומיטים, ספריות, קבצים) מ LR ל RR.

פקודה זו תיכנס לתוקף אך ורק אם:

* ה- head branch הוא RTB העוקב אחרי RB מתאים.
* ה- branch הרלבנטי ב RR נמצא **בדיוק** היכן שהוא מסומן על פי ה RB בLR .

**בונוסים שמומשו:**

1. **אנימציות**–

* איפוס branch מזיז אותו למיקום החדש על עץ הקומיטים, פונקציונאלית בלבד, אין אנימציה מיוחדת, אך כן רואים בבירור לאן ה-Branch עבר בעץ הקומיטים
* כשלוחצים על branch בעץ הקומיטים מדגישים את כל הקומיטים השייכים לו ב"פעימת לב"

מתחת לכפתור המאפשר לשנות את שם המשתמש, מוצג הכפתור 

הפעלתו תאפשר את הפעלת האנימציות, כל עוד הכפתור אינו לחוץ, האנימציות לא יוצגו.

1. **אפשרות החלפת Skin**

ישנן כמה ערכות צבעים שהמשתמש יוכל להחליף ביניהם

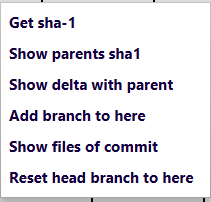
בצד בקצי השמאלה התחתון של האפליקציה מופיע כפתור

לחיצה עליו משנה את ה-Skin של האפליקציה.

החלפת ה-Skin כוללת החלפת רקע של המסך, החלפת המראה של הכפתורים, והחלפת הגופן של ה-Labal-ים המופיעים על המסך

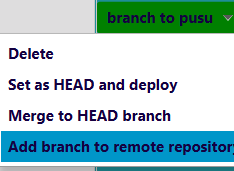
1. **תפריט הקשר לבחירה מתוך כל Commit בעץ קומיטים**

מתפריט ההקשר ניתן יהיה לבצע את הפעולות השונות בהקשר של קומיט/branch זה ביתר קלות:



1. **דחיפת Branhc חדש ל-RR**

מתוך תפריט ההקשר שברשימת ה-Branch-ים, תופיע האופציה לדחוף Branch ל-RR אך ורק במידה ומדובר ב-Branch חדש

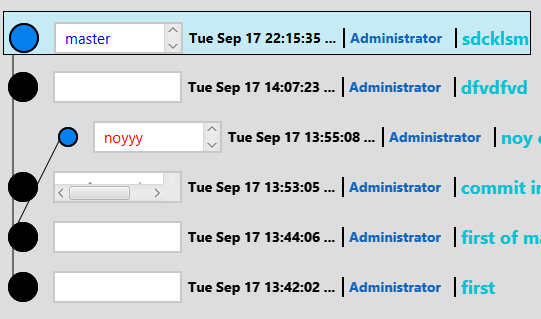


1. **גרף קומיטים איתן!:**

אלגוריתם ה layout שלנו לקומיטים השונים בעץ הקומיטים, הינו אלגוריתם שאנחנו הגינו בעצמינו, לפני שאביעד סיפק את שלו באחד השיעורים. האלגוריתם אינו נופל לא משנה כמה תאתגרו אותו, ובנוסף לפונציונאליות הרחבה של העץ (הפונקציונאליות רחבה אפילו יותר מכפי שנדרש בבונוסים), העץ עומד בהיר ונהיר!

1. **הדגשת Branch בעץ הקומיטים**

ברגע שהמתמש נעמד על Branch כלשהו במערכת, או אפילו על Commit כלשהו במערכת (!!) יוגדלו ויודגשו כל הקומיטים השייכים אליו (אבות ואבות קדומים)



1. **תמיכה בקומיטים יתומים ואלמנות**

בברירת המחדל עץ הקומיטים מכיל אך ורק את הקומיטים המוצבעים ע"י branc'ים בישירין או בעקיפין.

למשתמש יש האפשרות לבחור במצב תצוגה של 2 עצים שהוא יוכל לברור בינהם, באמצעות הכפתור המופיע ישר מעל לעץ הקומיטים 

1. עץ דיפולטי – כזה המציג **רק** את הקומיטים המוצבעים ע"י branch'ים בישירין או בעקיפין
2. עץ מלא – כזה המציג את **כל** הקומיטים, כולל היתומים מהצבעות מקור

**מחלקות עיקריות**

להלן המחלקות **העיקריות** בתוכנית (לפי Package):

**Magit-GUI.MainApp**

**MainAppController** – המחלקה האחראית על ה-GUI של התוכנית. מכאן מתחילה התוכנית (פונ' main). מחלקה זו אחראית על ניהול ה-GUI לרבות קשרים עם הקומפוננטות השונות על המסך הצגת הפלט למשתמש וקליטת הקלט מהמשתמש.

**Magit-Logic**

**MyAmazingGitEngine** – המחלקה האחראית על הלוגיקה של האפליקציה. מחלקה זו מכילה בתוכה מתודות Public המציעות את כל השירותים הנחוצים לאפליקיציה, כגון ביצוע commit, הצגת מצב מערכת הקבצים וכו'. לצורך ניהול ה היא שומרת בתוכה data members לצרכים הבאים:

* m\_ActiveRepository – ה-repository הנוכחי עליו git עובדת כרגע
* m\_UserName – שם המשתמש הנוכחי

**Repository**

מחלקה שאחראית על המידע של ה-repository הנוכחי, מחלקה זאת מכילה data members מטיפוס Magit, WC וכו' העוזרים לה לנהל repository כפי ש-git מנהלת