**MyBox System – Answers**

***GroupNum 1 – 23.6.15***

|  |  |
| --- | --- |
| Kostia Ilnitzki | Ikostia18@gmail.com |
| Simon Raviv | lionsimonlion@gmail.com |
| Gur Arye Yehuda | gury87@gmail.com |
| Eran Nahum | erann1987@gmail.com |

1. תארו בקצרה את תהליך קבלת ההחלטות והשיקולים שנקטתם בעיצוב תוכנת המערכת שפיתחתם. לוו את תשובתכם בתיאור דוגמאות ספציפיות מתוך המערכת שפיתחתם.

תהליך קבלת ההחלטות והשיקולים בעיצוב המערכת:

בתחילת המימוש הוגדרו ע"י חברי הקבוצה קווים מנחים לעיצוב וקווים כלליים לגבי מסמך הדרישות.

התקיימו דיונים לגבי הצגת הקבצים ב - WorkSpace. לאחר התלבטות בין שתי גישות, האחת טבלאית והשנייה בצורת רשת כפתורים, לשם תצוגה גרפית נעימה יותר לעין של מבנה הקבצים והתיקיות, הוחלט בסופו של דבר על רשת כפתורים.

בנוסף, הוחלט על תפריט שילווה את המשתמש לכל אורך הדרך בצדו של המסך, כך שיהיה תמיד זמין לביצוע פעולות.

הוחלט על ממשק פשוט וקל לתפעול של המשתמש, שניתן להתמצא בו ללא שום הכשרה מיוחדת. ממשק שיהיה מסוגל לענות על כלל הדרישות הרלוונטיות. ממשק אחיד כך שלא יטעה את המשתמש מהעבודה השוטפת. בממשקים הוכנסה מחשבה כך שכפתורים לא רלוונטיים ללחיצה לא מופעלים - יופעלו רק לאחר פעולה רלוונטית ובכך בעצם "מובילים" את המשתמש בדרך ללא טעויות לתפעול קל ויעיל.

דוגמאות לכך: למשל, עדכון הרשאות לקובץ קיים: לא ניתן לבחור קבוצה להרשאת "group" אלא רק לאחר שזו נבחרה כהרשאה ראשית.

כחלק מהאחידות נוצרה מחלקה אבסטרקטית שמכילה פעולות בסיסיות של כל חלון GUI על מנת שנוכל לדלג בקלות בין חלונות הGUI השונים.

בנוסף קיימת מתודה כללית שממומשת בכלל החלונות שמקבלת תשובה מהשרת ומטפלת בכל מקרה לגופו (הזדהות משתמש, יצירת קובץ, הודעה על יצירת קובץ שכבר קיים במערכת וכדומה).

1. תארו את תהליכי הבדיקות שביצעתם במהלך פיתוח הפרויקט שלכם. ציינו את המרכיבים המאפיינים של התהליך, בהתייחס לנלמד בהרצאות, ותוך מתן דוגמאות ספציפיות שביצעתם בפועל במהלך הפרויקט.

תהליך הבדיקות שביצענו במהלך פיתוח הפרויקט כלל את המרכיבים הבאים והתנהל בצורה הבאה:

במהלך כתיבת הקוד השתמשנו בשיטת Glass-box testing ע"י שיתוף פעולה בין חברי הצוות באופן כזה שכל חבר צוות שסיים מודול מסוים, הריץ בדיקות באופן עצמאי, תיקן את הנדרש ולאחר מכן העביר חלק זה לבדיקה לחברי הצוות האחרים שיבדקו פעם נוספת.

לרוב התבררו באגים קטנים שתוקנו ע"י מי שכתב את המודול.

כל זאת על מנת לוודא שאנו עומדים בדרישות ושעבור כל תהליך עם קלט מסוים נקבל את הפלט הרצוי.

על מנת לוודא תקינות התהליכים השתמשנו בשיטת Equivalence classes ע"י כך שחילקנו את המקרים למקרים כלליים ולמקרי קצה. כל אלו נבדקו בקפידה ובשלבים.

החלק האחרון של הבדיקות נעשה בשיטת Black-box, נעשו הרצות על גבי המערכת הסופית בקבצי ה-jar, במחשבי המכללה, נרשמו הערות אחרונות לתיקון.

1. התייחסו לאופן שבו התנהלתם לגבי 2 מרכיבים של הפרויקט:

א. ניהול הזמן ועבודת הצוות.

ב. שילוב קוד ובדיקות. אם היו קשיים, מה הסיבה לכך? מה הייתם משנים בדיעבד?

א.

נתייחס לגבי מרכיב ניהול הזמן וניהול עבודת הצוות בין 4 חברי הצוות.

כבר בתחילת הדרך של המטלה הראשונה ולאורך כל הפיתוח הוגדרו נהלי עבודה.

בכל קבלת מטלה התבצעה פגישה ודיון בין חברי הצוות, חולקו משימות בצורה ברורה, למען מיטוב עבודה מקבילית בין חברי הצוות ונקבע מועד אחרון לסיום המשימות הנוכחיות.

בכל סיום שלב כזה התבצעה פגישה מחודשת והוצגו הממצאים לאותו מועד.

העבודה האישית התבצעה לפי התכנון האישי של כל חבר צוות.

העבודה המשותפת והפגישות התקיימו לרוב בוועידות און ליין בסקייפ או בteam viewer.

ב.

שילוב הקוד, כמו גם שילוב הבדיקות התבצעו בצורה הבאה:

סביבת העבודה ושיתוף הקוד של הפרויקט התבצע בתוכנת ה-SmartGit בשרת של bit-bucket.

ההתנהלות בסביבה זו לא הייתה קלה ללימוד. בתחילה נתקלנו בקשיים רביים לביצוע סנכרון בין המחשבים האישיים לבין השרת ששומר את הקוד.

למרות זאת היה לנו חשוב להבין את התוכנה להמשך עבודה יעילה יותר.

כל חבר צוות שביצע עדכון בקוד שיתף אותו והודיע על כך לשאר החברים.