**שאלה 1:**

**סעיף א: תארו דילמות הנדסיות )issue Design )ספציפיות )לא כלליות( בהן עסקתם בתכן של ביצוע בחינות: ממוחשב וידני. תארו את השיקולים וקבלת ההחלטות שלכם והסבירו את הפתרונות שבחרתם.**

הדילמות שנתקלנו בהם הן:

1. בעת לקיחת מבחן ידני, מתי ליצור את הקובץ word – האם בעת בקשת הבחינה מהסטודנט, או בעת ההוצאה מהמגירה . החלטנו ליצור את הקובץ בעת הוצאתו מהמגירה. החלטה זו נבעה מכך שבמידה וזה היה מתבצע בעת בקשת בחינה ע"י הסטודנט השרת היה כל פעם מחדש יוצר את הקובץ -כלומר היו נוצרים העתקים של אותו הקובץ כמספר הסטודנטים. לכן החלטנו שבעת הוצאה מהמגירה זה יותר יעיל כיוון שהקובץ נוצר פעם אחת בלבד והדבר מוריד עומס מהשרת.
2. התלבטנו האם להפסיק את המבחן כשזמן נגמר בצד הclient או בצד הserver . בחרנו בצד השרת זו כיוון שדרך זו יותר בטוחה במובן הזה שאם הדבר היה בצד הלקוח סטודנט – הסטודנט יכול היה לשנות את הזמנים שלו.

**סעיף ב: ציינו איזו מהנושאים מההרצאות 6 - 10 בקורס באו לידי ביטוי בתהליך התכן המערכתי שביצעתם והסבירו באיזה אופן – באמצעות דוגמאות ספציפיות מתוך המערכת "AES.**

דפוסי התנהגות ודפוסי מבנה- באופן מבני הגדרנו את דרך הפעולה של שרת לקוח-כלומר הגדרנו את המבנה שבו יש לשלוח בקשות לשרת ואיך התשובות מצדו. לכן יש לנו מחלקה שהיא מייצגת חבליה(packet) שמכיל בתוכו את ההוראה שיש לבצע, את הdata הנדרש על מנת לבצע ואת הid של הלקוח שאליו צריכה לחזור התשובה. חבילות אלהה נוצרות בקונטרולרים של החלקים השונים שהם מהווים את הפעולות שניתן לבצע בכל חלק.

אינטגרציה- אצלנו בקבוצה הוחלט כי יש נציג אחד שאחראי על האינטגרציה. כל אחד עבד על המודולים שנדרש לפתח תוך הסכמה על הדרך הכללית שמתאימה לחברי הקבוצה ופעם בשבוע הועלו כל הקבצים שכבר סיימנו ויש להכניס. אותו הנציג היה מכנים ומעדכן את קובץ כך שהוא מוסיף את הפונקציונליות שהתווספה ובודק כי הדברים מתחברים כמון שצריך. בצורה זו תמיד הייתה לנו גרסה מעודכנת של התכנה שאפשר להמשיך לעבוד עליה.

**שאלה 2:**

**תארו את תהליכי הבדיקות השונים שבצעתם במהלך פיתוח הפרויקט שלכם. ציינו את מאפייני תהליכי הבדיקות תוך התייחסות לעקרונות שנלמדו בהרצאות, ותוך מתן דוגמאות ספציפיות שביצעתם )או לא ביצעתם( במהלך הפרויקט )ע"י תיאור מפורט של בדיקות מרכיבי מערכת "AES".**

תהליך היה כזה: כל אחד היה אחראי לבדוק את הקוד שכתב – כלומר האם הפונקציונליות באמת עובדת כמו שצריך- מה קורה במקרי קצה , האם הדברים כתובים בצורה שניתן לשלב אותה עם שאר הקוד.

לאחר מכן מי שהייתה אחראית על שילוב הקוד הנכתב -כשהיא יודעת שהוא כשלעצמו תקין-הייתה אחראית לבדוק האם השילוב שלא בקוד לא יצר רגרסיה מבחינת הפונקציונליות שכבר הייתה קיימת במערכת או לא ,ואיך הגרסה החדשה שנוצרה באמת מתנהגת. למצוא באגים התנגשויות וכיוצא באלה. מה שניתן היה לסדר לבד באחריות אותו האדם היה להעביר לחברי הקבוצה לתקן. השתדלנו לכתוב את התכנית בצורה שבה תהיה גמישות רבה ככל האפשר בחלוקת העבודה שתאפשר הכנסה קלה יותר ושילוב קל יותר של מודולים שעוד היו צריכים להיות מפותחים בעתיד. בנוסף השתדלנו לבדל באופן חד וברור את השרת והלקוח אחד מהשני באופן בוא בנינו אותם כך שיהיה ניתן לפתח את הבקשה ואת הטיפול פה באופן שהוא לא תלוי כמעט אחד בשנייה מכיוון שחילקנו בצורה כזו שתהיה נוחה לבדל את החלקים ככל שאפשר הדבר נתן אפשרות לבדוק את המודולים השונים ללא תלות האחד בשני .

**שאלה 3:**

**סעיף א: תיאום פעילויות ושיתוף בין חברי הצוות בפיתוח וגישה לניהול גרסאות: תארו את השיטה שלפיה פעלתם בהקשרים אלה, וציינו יתרונות וחסרונות שלה. יש להתייחס גם לתהליך העבודה - לא להתמקד רק בכלים ואספקטים טכניים.**

הגישה שבה עבדנו מבחינת ניהול גרסאות הייתה כזו: יש חבר צוות אחד שאחראי להיות תמיד המאחד. חילקנו את העבודה בין חברי הצוות ואחת לשבוע העלנו גרסאות מעודכנות של העבודה לgit . חבר הקבוצה שאחראי על האיחוד איחד את העבודות ובדק באופן וודאי שאין דברים מתנגשים. המטרה הייתה תמיד להיות עם גרסה של הפרויקט שעובדת ומאוחדת עד לאותה נקודת זמן כך שתמיד מצד אחד תהיה גרסה מעודכנת עם כמה שיותר פונקציונליות ומצד שני אם בעקבות שינוי מסוים נוצרה בעיה מאוד קריטית יהיה ניתן לחזור לגרסה אחת לפני הפונקציונליות שהכנסנו בגרסה האחרונה ובכך להיות במצב שבו תמיד יש לנו גרסה עובדת. באופן רפלקטיבי אפשר לומר כי שיטת עבודה זאת אמנם אפשרה פיתוח מהיר של המודולים השונים אבל הקשתה רבות על חבר הצוות שהיה צריך לחבר בין כל החלקים השונים ואף העמיסה עליו עד מאוד, בנוסף שיטה זו השאירה לסוף הרבה דברים שקשורים לביצוע פעולות שמחברות בין המודולים והיה קצת לחץ.

**סעיף ב: שילובי קוד אינטגרציה - לאחר הפיתוח הראשוני( ובדיקות. ציינו באופן פרטני, בהתייחס ספציפית לפיתוח המערכת "AES ," איך פעלתם בשלב זה של הפיתוח )למשל: תיאור התנהלות התהליך, אופן טיפול בבעיות, וכו'(. אם היו קשיים מה הסיבה לכך? מה הייתם משנים בדיעבד בגישתכם למרכיב זה מבחינת האספקטים הרלבנטיים של הנדסת תוכנה?**

תהליך העבודה שלנו היה כזה: נפגשנו על מנת להחליט מי עובד על איזה חלק, אחת לכמה זמן מאחדים מי מאחד מה יכלול באופן כללי הDB ומהי הדרך שבה אנו מצפים אחד מהשני לתכנת. לאחר ההסכמות הנ"ל התחיל תהליך המימוש. קבענו על יום בשבוע בו צריך להעלות את מה שנעשה לgit וגם על מנת להיות עם היד על הדופק ולראות מה מצבנו מבחינת המודולים השונים מי צריך עזרה ובאופן כללי האם היו בעיות. לאחר מפגש כזה היה מישהו בקבוצה שאיחד את כל הקבצים לכדי קובץ אחד עובד והוא שהעלה בחזרה לgit את הקובץ הזה על מנת שתהיה לנו גרסה טובה שכל הפונקציונליות בה עובדת עד אותה נקודת זמן.

בדיעבד ניתן לומר שהיה צריך לאחד את הקוד יחד כי מה שנוצר זה עומס מיותר על מי שאיחד את הכל מכיוון שהיה צריך גם לבדוק שכל החלקים שכרגע בגרסה משתלבים טוב ולשים לב איפה יש בעיות ולאתר אותם ואף לפעמים לפתור אותן בעצמו.