|  |  |
| --- | --- |
| **Group Number 10** | **12/06/2016** |
| Maisam Marjieh | [mar.m94@hotmail.com](mailto:mar.m94@hotmail.com) |
| Muhamad Igbaria | [muhamadig.ac@gmail.com](mailto:muhamadig.ac@gmail.com) |
| Bolous Abu-Jaber | [ajbolous@gmail.com](mailto:ajbolous@gmail.com) |
| Ahdab Serhan | [ms.ahdab@gmail.com](mailto:ms.ahdab@gmail.com) |
| Ahmad Mnasra | [ahmad.mnasra@gmail.com](mailto:ahmad.mnasra@gmail.com) |

שאלה 1 :

תארו את תהליך השיקולים וקבלת ההחלטות שלכם בתכן (design )המערכת לטיפול בזימון תור לרופא מומחה עבור המטופל. תארו דילמות הנדסיות (Design issue )ספציפיות לא כלליות מתוך תכן המערכת שפיתחתם בהקשר זה )בהתאם להסבר ולדוגמאות בהרצאה(, והסבירו את הפתרונות שבחרתם.?

בתכן המערכת לזימון תורים השתמשנו בישויות עבור רופא שמוגדר לו באיזה מרפאה הוא עובד ומה ההתמחות שלו , בנוסף לישות של תור שבו מצוין בין היתר את הרופא ,המטופל,מועד התור,תאריך יצירת התור .

המשתמש(מוקדן) עליו לבחור קודם את ההתמחות שהמטופל מבקש תור עבורה , המערכת דואגת להציג לו את כל ההתמחויות האפשריות בכל המרפאות , עם בחירת התמחות מסוימת המערכת דואגת לבדוק אם למטופל יש אישור רפואי עבור התמחות זו , אם לא היא לא תאפשר המשך התהליך , ואם יש אישור אז היא מציגה את כל הרופאים בעלי אותה התמחות בכל מרפאות הרשת ממוינים לפי זמן הביקור האחרון של המטופל אצל הרופא .

עבור כל רופא מוגדר משמרות ,שהטיפול בקביעתן נעשה באמצעות מערכת חיצונית , כל משמרת מכילה את הרופא , תאריך ושעת התחלה וסיום.

כאשר המוקדן בוחר רופא מסוים המערכת תציג טבלה של מועדים אפשריים בהסתמך על המשמרות שנקבעו עבור הרופא כך שלא תהיה חפיפה בין המועדים האפשריים לתורים שנקבעו אצל הרופא וגם אצל המטופל ,במקרה שאין לרופא הנבחר משמרות מוגדרות המערכת מדווחת .

המוקדן בוחר את המועד המבוקש ומוסיף את התור שמתעדכן כתור של המטופל והרופא בטבלת התורים.

דילמות:

* היינו צריכים להחליט האם להציג רשימה קבועה של התמחויות ללא קשר אם קיימת התמחות כזו באחת המרפאות או לא , החלטנו שהרשימה תהיה מעודכנת כל הזמן באמצעות שאלתה שמקבלת את ההתמחויות מטבלת הרופאים .
* אחת הבעיות הייתה איך בעצם נציג את טבלת הרופאים ממוינת לפי הביקור האחרון , אשר בעצם לא ניתן לשמור שדה במסד הנתונים , במקרה זה השתמשנו בשאלתה שתחזיר עבור כל רופא ברשימה את מועד הביקור האחרון של המטופל אצלו ואז למיין את הטבלה.
* בהצגת מועדי זימון התור האפשריים היינו צריכים להגדיר עבור כל רופא את שעות העבודה שלו (משמרות) , היינו צריכים להחליט איך המשמרות מוגדרות , האם באופן קבוע עבור תקופה מסוימת , או הגדרת המשמרות באופן אינדיבידואלי , החלטנו להגדיר את המשמרות באופן אינדיבידואלי כל משמרת לחוד עם תאריך ושעת התחלה וסיום .
* בנוסף לכך מה הזמן ששוהה המטופל אצל הרופא , חלקנו את מועדי התורים ל (slots) של 30 דקות בין כל תור לאחר.
* סנכרון בזמן אמת: אחת הבעיות שהמערכת עלולה לפגוש היא קביעת תור באותו רגע לאותו מועד ולאותו רופא ע"י שני מוקדנים שונים , ניסינו לייצור בדיקה אחרונה לפני שמירת התור במסד הנתונים ולוודא שהתור הזה עדיין פנוי.

שאלה 2 :

תארו את תהליכי הבדיקות שבצעתם במהלך פיתוח הפרויקט שלכם. ציינו את המרכיבים המאפיינים של התהליכים, בהתייחס לנלמד בהרצאות, ותוך מתן דוגמאות ספציפיות וקונקרטיות שביצעתם(או לא ביצעתם) בפועל במהלך הפרויקט ( בהקשר של מרכיבי המערכת הספציפיים(.

תשובה :

* בהתחלה בדקנו את המערכת בהדרגתיות ,ז"א שבדקנו כל תת מערכת בפני עצמה שהיא תקינה ועונה על הדרישות למשל בדקנו את ההתחברות של ה Server עם ה Database ושבאפשרותו לבצע שאילתות מול ה database ובדקנו דרך טבלאות ה database את תוצאות השאילתות אם נותנות את התוצאה הצפויה .
* בשלב מתקדם בכתיבת הקוד בדקנו את המערכת באופן vertically ז"א בדקנו כל תהליך מהפעולה שמבצע המשתמש עד ביצוע השאילתות ב DataBase
* חלוקת הבדיקות לקטגוריות למשל אחת הביות שטפלנו היא בקטגורית ordinary algorithm של Off-by-one errors כאשר תוצאות השאילתות נשמרות ב arrayList / Collection ואז במקום לעבור על האיברים באמצעות לולאת for רגילה השתמשנו בלולאה for מקוצרת או ב iterator כדי למנוע חריגות .
* בבדיקת תקלות מסוג Incorrect logical conditions חלקנו את המקרים שיש לבדוק למחלקות שקילות ואז בדקנו את מקרי הקצה (בדקנו ערכים נכונים וערכים לא נכונים),למשל בתהליך ביטול תור בדיקת אם תאריך התור הוא בתאריך היום אז לא ניתן לבטל התור ואז כדי לבדוק תנאי זה נסינו לבטל תור שהוא בתאריך היום ותור אחר שהוא לא בתאריך היום (שניתן לבטל אותו ).
* מניעת טעיות : בממשק המשתמש נסינו כמה שיותר למנוע מהמשתמש להכניס ערכים לא נכונים ,למשל בזימון תור הצגנו למוקדן רק את ההתמחויות שקימות בפועל במרפאות הרשת וכך מנענו ממנו לבחור התמחות שלא קיים אף רופא בהתמחות זו .וגם בהצגת המועדים האפשריים בקביעת תור מנענו בחירת מועד שתפוס (מהרופא או מהמטופל) .

שאלה 3 :

תחקור והפקת לקחים: התייחסו לאופן שבו התנהלתם לגבי 2 מרכיבים של הפרויקט: (א) שיתוף בין חברי הצוות בפיתוח וניהול גרסאות: ציינו יתרונות וחסרונות של השיטה שלפיה פעלתם(יש להתייחס לתהליך - לא להתמקד רק באספקטים טכניים). (ב) שילובי קוד (לאחר הפיתוח – אינטגרציה)ובדיקות. אם היו קשיים מה הסיבה לכך? מה הייתם משנים בדיעבד?

תשובה :

א. בהתחלה בנינו את ה prototype של התכנית ואת ה Entities לפי תרשים המחלקות וקבענו את תכן המערכת כך שחלקנו הפרויקט לשלושת תת מערכות שמיישמות שיטת שלוש השכבות ואז חלקנו את המשימות בין חברי הקבוצה לפי use case כך שכל חבר קבוצה עבד באופן מקביל על use case מסוים באופן מלא (מממשק המשתמש עד ביצוע הפעולה במסד הנתונים ) .

יתרונות של שיטה זו ההתקדמות במימוש המערכת באופן מהיר וממוקד .

חסרון השיטה שיש תלויות בין ה use cases השונים דורש סנכרון בין חברי הצוות וזה עלול לעקב את ההתקדמות ..

ב. במהלך כתיבת הקוד השתמשנו ב GitHub, כך שעשינו שילוב של הקודים כל הזמן , בנוסף לפגישות ביניים בין חברי הקבוצה לסיכום מה שנעשה עד עכשיו.

בסוף הפיתוח עברנו כל חברי הצוות על התוכנית ובדקנו אותה , תוך כדי תיקונים של בעיות ובגים.