מסמך אפיון פרויקט

פרויקט מעשי לתואר B.Sc במדעי המחשב

Electify

חוזה חכם לבחירות חכמות מבוססות בלוקצ'יין וקריפטוגרפיה

חברי הצוות:

עינבר ישראלי - 205925290
הדס ישראלי - 207041443
אוהד אדרי - 205533003

מרצה מנחה – מר רועי זימון חתימה _____

תאריך הגשה – 5.9.2021

הצעת פרויקט בפקולטה למדעים

מכון טכנולוגי חולון- H.I.T.

שימו לב! יש למלא את הטופס באופן דיגיטלי. אין לשנות מבנה הטופס.

טופס סרוק או עם פרטים חסרים לא יתקבל.

סמן ב **X** את מספר הקורס:	**X**	מדעי המחשב	1	0	0	6	6
		מתמטיקה שימושית	6	0	2	1	2

שנה עברית			
ת	ש	פ	א

סמסטר תחילה:
ב

שנתי/סמסטריאלי:
שנתי

תאריך הגשת ההצעה (סטודנט): 12/02/2021 | **סוג פרויקט (תעשייתי/מחקרי):** תעשייתי

א. פרטי הסטודנט/ים:

שימו לב! ניתן לקחת פרויקט שנתי רק פעם אחת במהלך כל תואר.

	שם מלא	**אוהד יהודה אדרי**
	מחלקה	**מדעים**
	תעודת זהות	**205533003**
	טלפון	**050-2035343**
	דואר אלקטרוני	**edryohad@gmail.com**

	שם מלא	**הדס ישראלי**
	מחלקה	**מדעים**
	תעודת זהות	**207041443**
	טלפון	**052-6559715**
	דואר אלקטרוני	**hadas25496@gmail.com**

חתימת הסטודנט

אוהד אדרי

_____

חתימת הסטודנט

הדס ישראלי

_____

שימו לב! ניתן לקחת פרויקט שנתי רק פעם אחת במהלך כל תואר.

	שם מלא	עינבר ישראלי
	מחלקה	מדעים
	תעודת זהות	205925290
	טלפון	052-6559715
	דואר אלקטרוני	israeli.inbar@gmail.com

	שם מלא	
	מחלקה	
	תעודת זהות	
	טלפון	
	דואר אלקטרוני	

חתימת הסטודנט

עינבר ישראלי

_____

ב. שם הפרויקט

שם הפרויקט בעברית:

פינטק- חוזים חכמים מבוססי בלוקצ'יין

Project Name in English:

Fintech - Block Chain

ג. שמות המנחים:

מנחה אקדמי: _____ תאריך: _____ חתימה: _____

מנחה אקדמי/תעשייתי: _____ תאריך: _____ חתימה: _____

ד. רקע לפרויקט. תיאור הבעיה

לספר מהי הבעיה שהובילה לפיתוח הפרויקט ומהי תרומתו של הפרויקט לתחום בו עוסק.

חוזים חכמים (מבוססים בלוקצ'יין וקריפטוגרפיה) מאפשרים לנו לבצע העברות כספים, נכסים, מניות וכל סוג ערך אחר בצורה שקופה, ללא קונפליקטים או בעיות, תוך ביטול הצורך בשירותי תיווך עבור העסקה. ניתן להסתכל על חוזה חכם כאל מכונת פחיות. אם ברצונכם לרכוש פחית קולה ממכונת הפחיות במשרד שלכם – אתם תיגשו למכונה, תכניסו כסף, תלחצו על כפתור הקולה והמכונה תביא לכם את הפחית. חוזה חכם עובד בצורה דומה של "אם... אז...". עם חוזים חכמים, תוכלו לשלוח את הביטקוין / את'ר שלכם לכתובת הארנק של חוזה חכם כלשהו, ולקבל את התמורה בצורה אוטומטית לפי הקוד של החוזה. חוזים חכמים לא רק יוכלו להגדיר את החוקים, תנאים והעונשים הסובבים הסכם כלשהו, כפי שנעשה כיום בהסכמי נייר – אלא גם להוציא עצמם אוטומטית לפועל בהתאם לטריגר שהוזן מראש.

לשם הבנה ברורה יותר, אתן דוגמה: כיום, אם אנו רוצים להזמין מונית, ככל הנראה שניצור קשר עם .Get Taxi גט טקסי היא מתווך – היא הערבות שלנו כי מדובר בנהג מונית אמין, והיא הערבות של הנהג לכך שלא נברח לו ללא תשלום. כעת, אצור חוזה חכם ואזין אותו בלוקצ'יין, ואקרא לו .Taxi Contract שני הצדדים המעורבים בחוזה הם נהג המונית ואני, הצרכן. הקוד של החוזה החכם פתוח, כך ששני הצדדים יוכלו לקרוא את כל החוקים, תנאים ועונשים בחוזה. הטריגר שהזנתי לחוזה החכם הוא התקדמות בקילומטרים. זאת אומרת, החוזה יידע לתקשר עם קילומטראז' הרכב (לצורך הפשטות נניח שזהו פיצ'ר שכבר קיים כיום), וברגע שנסענו קילומטר במונית – יישלחו 3 מטבעות קריפטוגרפים מהארנק שלי לארנק של נהג המונית. אם נסעתי 20 קילומטר, יישלחו לו 60 מטבעות קריפטוגרפים. כפי שכבר הבנתם, החוזה החכם מגשר על הפערים, שעד כה Get Taxi גישרו.

ה. מטרות הפרויקט

להסביר מהן המטרות אותן רוצים להשיג בפרויקט זה ומהם תוצרי הפרויקט שיסופקו

פיתוח מערכת מאובטחת לחוזים חכמים (מבוססים בלוקצ'יין וקריפטוגרפיה) המאפשרת לבצע פעילות עסקית מאובטחת כגון: העברות כספים, נכסים, מניות וכל סוג ערך אחר בצורה שקופה, ללא קונפליקטים או בעיות, בין עמית לעמית, תוך ביטול הצורך בשירותי תיווך עבור העסקה של גורם שלישי כמו בנקים. חוזים חכמים יוכלו לא רק להגדיר את החוקים, התנאים והעונשים הסובבים הסכם כלשהו, אלא גם להוציא עצמם אוטומטית לפועל בהתאם להסכמים שהוזנו מראש..

ו. הגדרת דרישות פיתוח

פרט את מדדי ביצוע בהם יבחן הפרויקט , להבטיח כי עמד בהצלחה מול דרישות פיתוח

פיתוח סוכנים חכמים באופן מבוזר - מטרת הסוכנים הינם לייצר בלוק שנקרא חוזה עם פרטי העסקה כגון : שם בעל הנכס , פרטי הנכס , תאריך ההנפקה , פרטי מכירה

פיתוח שרת לניהול הסוכנים בצורה חכמה - תפקיד השרת הינו לוודא כי כל פעולה של ניהול החוזים אשר נוצרים על ידי הסוכנים החכמים , נבדקים ועוברים בדיקות אבטחה כגון זיהוי בעל הנכס , וידוא כי לא נוצר אובייקט משוכפל עם פרטים שקיימים כבר בהיסטוריה הרכישות, וזאת באמצעות שימוש בטכנולוגיות **לדג'ר – יומן דיגיטלי היושב על רשת מבוזרת.**

טרנזקציות מאובטחות, יוצרו על ידי שימוש בפונקציות הצפנה וערבול במגוון אלגוריתמים קריפטוגרפים. התקשורת בין הסוכנים ובין שרת הניהול יתבצע באמצעות תעבורה מוצפנת ומאובטחת .

ז. משימות עיקריות ותוצרי פרויקט

פרט משימות עיקריות וכן סיכוני פיתוח (באם ישנן) להשגת מטרות הפרויקט

הגשת מסמך תכנון פרויקט. קובץ בפורמט.Word

מסמך מסכם , קובץ PDF וקובץ Word עם חתימות המנחים

פוסטר הפרויקט. קובץ PDF וקובץ PPTX בהתאם לתבנית מוגדרת

מצגת הפרויקט. קובץ PDF וקובץ.PPTX

סרטון לפרויקט. יוגש בפורמט MP4 ברזולוציה של.(FHD (1080p

קוד תוכנה אלגוריתמים ו/או תכנה (אב טיפוס ממוחשב עובד).

רישום תוצרי פרויקט בבסיס הנתונים באתר הפרויקטים.

שם רכז פרויקטים	תאריך	חתימה

תוכן עניינים

תקציר הפרויקט

פרויקט "Electify" הינו פרויקט מעשי דו סמסטריאלי בהנחיית המנחה רועי זימון ששם במטרתו להגדיר מחדש את הדרך בה אנו מצביעים בבחירות במדינה.

בשנים האחרונות תחום הבלוקצ'יין הפך לאחד הדברים החמים בעולם כיום. הרעיון המרכזי מאחורי בלוקצ'יין הוא יצירת מערכת לניהול חוזים חכמים ללא גורם מתווך.
את הגורם המנהל, מחליפים "בלוקים" מוצפנים של מידע הנוצרים בעזרת שיתוף מבוסס רשת תקשורת מסוג P2P (Peer to Peer).

הבלוקצ'יין מנוהל באופן מבוזר, ע"י נתונים שנכתבים אחת ליחידת זמן לתוך בלוק.
בתום זמן מסוים, ננעל הבלוק, והוא מצטרף כמעין שרשרת לבלוקים הקודמים.
הבלוקים מוצפנים (עוברים תהליך גיבוב) באמצעות שילוב של מפתחות פרטיים ומפתחות ציבוריים.

לשם כך פיתחנו את המערכת שנקראת "Electify" , וכשמה כן היא, מערכת המאפשרת בחירות חכמות על בסיס תשתיות הבלוקצ'יין והקריפטוגרפיה.

במסגרת הפרויקט, נשתמש בתשתית הקוד הפתוח הקיים על מנת לעשות דיגיטציה תוך שמירה על "טוהר הבחירות" ברמה המחמירה ביותר להליך הבחירות המיושן, הן בכנסת ישראל, הן ברשויות והערים המקומיות ועוד.

למעבר בין תהליך בחירות מיושן לחדש ומתקדם יש יתרונות רבים עליהם נפרט בהמשך הפרויקט אך יחד עם זאת גם קשיים, אך אנו מקווים להראות פוטנציאל בו השלם גדול מסך חלקיו.

אמנם אנו רק צוות קטן והפרויקט שלנו נחשב בגדר רעיון דמיוני נכון לכתיבת שורות אלו, באופן יחסי לפרויקטים "פרקטים" אחרים שיכולנו לקחת באותו נושא, אך אין לנו רצון להישאר באזור הנוחות אלא לפרוץ קדימה מבחינה רעיונית ומחשבתית. מי יודע, אולי בעתיד הרעיון הדמיוני יהפוך למציאות ונוכל להשפיע על מערכת הבחירות הבאה במדינת ישראל.

פרק מבוא

רקע

עם התקדמות המחשוב והאוטומציה בעולם באספקטים רבים של החיים, הוחל בניסיון למחשב הצעות שונות, בכמה אופנים, למשל: הצבעות במשחקי טלוויזיה, בחירות מקדימות של מפלגות ופרלמנטים שונים כמו הכנסת בישראל.

האם בעתיד הבחירות לכנסת ולרשויות המקומיות יתקיימו בשיטה אלקטרונית? במרבית המדינות בעולם ההצבעה היא עדיין ידנית, ויש מדינות שהטמיעו טכנולוגיות בתהליך ההצבעה אך נסוגו. עם זאת, מומחים לבלוקצ'יין סבורים כי זאת תהיה הטכנולוגיה שתפתור את הבעיות בהצבעה הממוחשבת.

מאז קום המדינה ועד היום, כל מערכות הבחירות בארץ, בין אם לכנסת ישראל או לרשויות המקומיות והערים המוניציפליות נעשו, ועדיין נעשים בדרכים מיושנות ע"י הצבעה פיזית בקלפי.

אזרח פלוני אלמוני מגיע לקלפי, נעמד בפני פאנל רחב של עובדי קלפי (מזכיר קלפי, סגן מזכיר קלפי, מפקח טוהר בחירות וכו'), האזרח מציג תעודת זיהוי, נעמד מאחורי פרגוד חסוי, בוחר פתק ומשלשל לתוך התיבה.

השעות החוקיות להצבעה בארץ הן בין השעות 7:00-22:00, בשעה 22:00 כל הקלפיות ברחבי הארץ נסגרות להצבעה. משעה זו מתחיל מרוץ נגד הזמן לספירת קולות וסגירת הקלפי. אחרי ספירת תוצאות האמת בקלפי, יו"ר הקלפי נוסע למרחב אזורי שאליו מתנקזים עוד יו"ר קלפי מערים/יישובים סמוכים. למקום זה מביאים את תוצאות האמת, כל יו"ר מאזור ההצבעה שלו, וכך הלאה בשרשרת אימות הנתונים ותוצאות האמת.
חשבו לנו לציין שפסקה זו, נכתבה מחוויה אישית של אחד מחברי הצוות בבחירות לכנסת ה-24.

בנוסף, תקופת הקורונה קראה תיגר חדש לדרך בא אנו מצביעים וכתוצאה מכך המדינה נאלצה לפתוח עוד עשרות קלפיות (שחלקן נפתחו אך ורק בשביל מספר חולים מצומצם והצורך לקיים את זכות הייסוד - הזכות לבחור ולהיבחר). בנוסף התווספו עשרות אלפי מעטפות כפולות (למעלה מחצי מיליון בבחירות 2021), דבר שלקח המון זמן ומשאבים לספירת קולות עד להגעת תוצאות האמת.

באופן כללי, בחירות בעולם ובאופן ספציפי בישראל עולות למשלם המיסים סכומי עתק (בישראל הסכום מגיע לבין 2-3 מיליארד שקל) בנוסף לסיבה שבשנים האחרונות בישראל, בחירות מוגדרות כיום שבתון – על מנת לעודד הצבעה ולמעשה תשלומי השכר לעובדים ביום שבתון הוא כ-200 אחוז על פי חוק כולל עובדי הקלפי שמתוגמלים בצורה נאה.

מטרות הפרויקט

Electify הינה פלטפורמה בתחום הבלוקצ'יין והחוזים החכמים. שימוש בבלוקצ'יין הינו נפוץ (NFT marketplaces, Cryptocurrency exchange, Real estate processing platform, Supply chain and logistics monitoring, etc.). ונהפך לשימוש פשוט ע"י ארנקים קרים/חמים לעשיית טרנזקציות בין אנשים.

אנו רוצים לחשוף את קברניטי המדינה לתחום הבלוקצ'יין והחוזים החכמים כדי להראות ולהוכיח שניתן לעשות שינוי באופי בו מתבצעות בחירות כיום.

בסיום הפרויקט ולאחר הסבר מקיף של המערכת לאנשים הרלוונטיים, הם יקבלו את הרושם בין אם בצורה חיובית (בשאיפה), או שלילית ,האם אנחנו כמדינה בשלים לעשות מעבר דיגיטלי בתחום רגיש שכזה.
למעשה, הלקוחות הפוטנציאליים שלנו הן ממשלות, ערים ויישובים מוניציפליים, משאלי עם שמאז ומעולם לא התרחשו בישראל בגלל מורכבות העניין.

הגדרת יעדים:
- בסיום הפרויקט, המשך פיתוח המערכת ואפיון עם גורמים בכירים בתעשייה.
- הצגה לאנשי מפתח בכנסת ישראל לצורך מחקר ויצירת עניין.
- התחלת פיילוט בהצבעות קטנות ומוגדרות מראש, תוך למידה לצורך שיפור המערכת.
- הצבעה בבחירות מוניציפליות כחלק מהצגת תכלית ביישובים/ערים ספציפיים.
- הצבעה בבחירות לכנסת ישראל כחלק מהצגת תכלית רחבה ביישובים/ערים ספציפיים.
- מעבר מלא להצבעה דיגיטלית תוך כ-10-8 שנים מהיום.

מה יש כיום? –

בנובמבר 2007 הרשויות המקומיות בישראל ביצעו פיילוט לבחירות ממוחשבות ב-11 קלפיות, ישנה כוונה לעבור בשלב מסוים להצבעה ממוחשבת ברשויות המקומיות, ותזכיר לחוק בנושא הונח על שולחן הכנסת לדיון מוקדם במאי 2009. בתזכיר זה מפורט תהליך ההצבעה, ובו נכתב כי לאחר זיהויו של המצביע וזכאותו להצבעה יונפק לו כרטיס חכם באמצעותו יצביע בעמדת מחשב באמצעות מסך מגע, בחירתו תירשם הן במחשב ההצבעה והן בכרטיס החכם, בסיומה של ההצבעה הכרטיס ישולשל לקלפי.

בנוסף, בספטמבר 2019 בעיר מוסקבה שברוסיה נעשה שימוש ראשון במערכת הצבעה אלקטרונית אבל מדובר בשימוש חלקי וניסיוני בלבד. המערכת, שאכן התבססה על בלוקצ'יין, נועדה להקל על ההצבעה ולהפוך אותה לנגישה יותר, בתקווה להגדיל את שיעורי ההצבעה.

יש הטוענים כי רצוי לעבור להצבעה ממוחשבת בבחירות לכנסת בהקדם האפשרי ,וקיימות אף שיטות שמטרתן למנוע אפשרות של זיופים, כולל בידי מתכנני מערכות ההצבעה. לעומת גישה זו, ישנם חוקרים מהאקדמיה ומהתעשייה שחוששים מאוד מתהליך של הצבעה ממוחשבת וטוענים שיש להתקדם לקראת הליך זה בזהירות מרובה ,שכן בעיה או תקלה רצינית בהליך זה, תרוקן את הדמוקרטיה מתוכן.

יתרונות של הצבעות דיגיטליות:

- צמצום טעויות הבוחרים, שיוכלו לבחור רק אחת מהאופציות העומדות לרשותם (ומתן אפשרות לבחירה של אנאלפביתים ,שיוכלו לזהות מועמד לפי תמונתו).

- מיידיות של התוצאה הסופית (וביטול סקרי הצבעה בצידן).

- הצבעה בכל מקום בשל ספר בוחרים ממוחשב (מה שעשוי להעלות את מספר הבוחרים) , וביטול הצורך להשתמש בהצבעה במעטפות כפולות (מה שיבטל את הזמן הנוסף של עיכוב בתוצאות האמת).

- ספירה מדויקת של ההצבעות והקטנה של השגיאות, הטעויות והזיופים בעת הספירה.

- חיסכון בחומרי הצבעה מתכלים כמו פנקסי בוחרים, מעטפות ופתקי הצבעה (אם כי בטווח הקצר החיסכון בטל בששים לעומת העלות של מכונות ההצבעה והשימוש בטכנאים).

חסרונות של הצבעות דיגיטליות:

- חוסר אינטואיטיביות וסיבוך לעומת הצבעה מסורתית (שעלול לגרום לזמן הצבעה ארוך מהרגיל ואף לתוצאות בחירה מוטעות).

- עלות ראשונית יקרה בהצטיידות במחשבים או מכונות הצבעה (העלות נאמדת לפחות פי עשרה מהצבעה ידנית).

- אפשרות של זיופים בידי כותבי מערכת ההצבעה ומפתחיה שיטו את תוצאות ההצבעה באופן דרסטי בקוד התכנותי עצמו או במחשבים המרכזיים (דבר שקשה הרבה יותר לבקר ולנטר מהצבעה בתהליך ידני).

- חשיפה להאקרים ואף טכנאים האחראיים שעלולים לנסות לפרוץ לעמדות ההצבעה ולשנות אותם או לשתול בהם וירוסים או לשבש את תוצאות הבחירות.

- רגישות המערכת לתקלות שונות כמו תקלות טכניות בעמדות, ניתוקים ברשת התקשורת, הפסקה בזרם החשמל ועוד.

כמו שניתן להבין מהכתוב, עולה חשש אמיתי מהצבעות דיגיטליות מסיבות שונות ובעיקר אבטחתיות, הרעיון שאנו מציעים בפרויקט מתבסס על בלוקצ'יין וחוזים חכמים, דבר שמעלה באופן משמעותי את השקיפות והאבטחה להצבעה בטוחה.

תכנון וביצוע הפרויקט

לדג'ר

ארנק חומרה קר (אינו מחובר לאינטרנט) אשר יוצר במיוחד לאבטחת המטבעות הדיגיטליים. מדובר בכונן קשיח חיצוני בעל הצפנה ברמה הגבוהה ביותר, שמאפשרת לו להיות חסין לכל סוגי ההתקפות, הארנק תומך במגוון רחב של מטבעות. לארנק זה יש יתרון והוא קישוריות בלוטוס אך החיסרון שלו די מורגש וזה כמובן פתח נוסף לנוזקות וחדירות.

הצפנה סימטרית

הצפנה סימטרית נקראת כך מכיוון שנדרש מפתח משני הצדדים. מדובר באלגוריתם הצפנה שבו משתמשים במפתח הצפנה **יחיד** הן להצפנה של הטקסט הן לפענוח טקסט מוצפן. צופן סימטרי מקבל טקסט קריא ומפתח הצפנה ובעזרתו ממיר את הטקסט הקריא לטקסט מוצפן שאינו מובן ואותו הוא שולח ליעדו. בצד המקבל, אלגוריתם הפענוח מבצע את הפעולה ההפוכה. כדי שהפענוח יצליח המפענח חייב להחזיק במפתח פענוח מתאים שמאפשר את הפיכת פעולת ההצפנה. הדרישה שיהיה מפתח אחד משותף לשולח והמקבל היא התכונה העיקרית המבדילה בין ההצפנות.

הצפנה א-סימטרית

הצפנה זו מכונה "א-סימטרית" מכיוון שביטחון שיטת המפתח הציבורי נשען על הקושי שבחישוב המפתח הפרטי מתוך המפתח הציבורי, שבניגוד להצפנה סימטרית בה מפתח הפענוח זהה למפתח ההצפנה. בהצפנה זו מפתח ההצפנה שונה ממפתח הפענוח, כלומר - כל משתמש מכין לעצמו זוג מפתחות: מפתח ציבורי (מפתח הנגיש לכולם) ומפתח פרטי (נשמר בסוד ומשמש לפענוח) ההתאמה בין המפתחות היא חח"ע, כלומר- לכל מפתח ציבורי קיים אך ורק מפתח פרטי יחיד המתאים לו וההפך.

כדי להצפין בשיטה זו על השולח להשיג עותק של המפתח של הציבורי של המקבל, רק המקבל יוכל לשחזר את הטקסט המוצפן בעזרת המפתח הפרטי המתאים שנמצא ברשותו. השימושים בהצפנה זו הן: הצפנת מסרים, חתימה דיגיטלית ואימות מסרים.

Hash Table – טבלאות גיבוב

מבנה נתונים מילוני שנותן גישה לרשומה באמצעות המפתח המתאים לה, המבנה עובד באמצעות הפיכת המפתח על ידי פונקציית גיבוב למספר המייצג אינדקס במערך שמפנה אל הרשומה המבוקשת. בעזרת טבלאות אלו, אנחנו יכולים לשמור מידע ולאחזר אותו בזמן קבוע.

בעיות במהלך תכנון ביצוע והפעלה

רעיון חדשני –

חדשנות היא יצירת ערך מוסף באמצעות פתרונות לצרכים גלויים, או באמצעות יצירת פתרונות לצרכים סמויים. המצאה היא פתרון חדש ובלתי צפוי לבעיה או קושי בעולם המוחשי, הנהגה במחשבתו של אדם.

חדשנות שונה מהמצאה, בכך שחדשנות מתייחסת לשימוש בשיטה או ברעיון טובים יותר, בעוד שהמצאה מתייחסת לתהליך ולאופן היצירה של השיטה או הרעיון.

כשחשבנו לאיזה כיוון נרצה לקחת את הפרויקט, רצינו לעשות משהו שונה וחדשני. בשיא המשבר הפוליטי שידעה אי פעם מדינת ישראל, חשבנו לעצמנו איך עדיין אין הצבעות דיגיטליות בישראל?. לקחנו נושא מיושן, לקחנו טכנולוגיה חדשה שתתן מענה לנושא האבטחה (בלוקצ'יין וקריפטוגרפיה) וניסינו לאחד בניהם. הקושי נבע מהסיבה שאומנם יש הצבעות דיגיטליות בחלק מהמדינות אבל לא על בסיס בלוקצ'יין וחוזים חכמים (למעט ניסיון קטן במוסקבה שברוסיה), לכן היינו צריכים לחשוב מהבסיס איך אנחנו רוצים שהמערכת תראה, ולא "להתלבש" על רעיונות קיימים כמו חוזי נדל"ן חכמים.

טכנולוגיות חדשות –

הפרויקט היה אתגר גדול עבורנו מהסיבה שאף אחד מחברי הצוות, חוץ מקצת אקטואליה על מטבע הביטקוין וקריפטוגרפיה, לא באמת הכרנו או ידענו מה זה בלוקצ'יין. נאלצנו ימים כלילות לחקור מאמרים וכתבי עת, לקרוא ספרים וכתבות, לראות סרטונים וסרטים ולהבין איך הטכנולוגיה עובדת. בנוסף השפות בהן השתמשנו לפיתוח הפלטפורמה חדשות לנו, וגם אותן נאלצנו ללמוד תוך כדי הפרויקט.

חיבור בין Frontend & Backend –

בתחילת העבודה על הפרויקט פיתחנו את החלק של האתר, Frontend. לאחר שסיימנו את רוב החלקים של ה-Backend, רצינו לחבר בין שני החלקים על מנת שנוכל להריץ את האפליקציה ולהמשיך לפתח אותה כפי שתכננו.

- נתקענו במיזוג של התיקיות, נאלצנו למצוא פתרון יצירתי ולכן יצרנו תיקייה של React-App בתוך החלק של הקוד של ה-Backend ולשם "להעתיק" את החלק של ה-Frontend ולהמשיך לפתח משם.

- שנתקלנו בה הוא למשוך ולקבל את הנתונים מה-localhost3000 אל האפליקציה - הפתרון היה לתת הרשאה מתוך השרת למשיכת נתונים לפורטים מסוימים, ולכן, עשינו הרשאה לכל הפורטים.

- פתיחת האתר והשרת באותו פורט localhost3000 - בעזרת Parcel הצלחנו למזג אותם לפורט אחד lochalhost:3000. בנוסף, בבעיה הקודמת של פתיחת האתר מפורטים נוספים שמאזינים לשרת במקום לתת הרשאה לכולם, השימוש ב-Parcel פתר גם את הבעיה הזו.

הבאת נתונים -

בעיה נוספת שנתקלנו בה היא הבאת נתוני בחירת המפלגה מתוך הבלוקים בבלוקצ'יין - בעזרת fetch הצלחנו להביא את הנתונים מתוך API/Blocks שמרנו את הנתון הרצוי כמערך מחרוזות מתוך json שחזר לנו.

הרשאות משתמשים –

ישנו מסך כניסה לאתר הכולל שם, ת.ז וסיסמא. בעיה - לא הוגדרו הרשאות למשתמשים המתחברים לאפליקציה. כל משתמש השמור במערכת יכול להיכנס לכל עמוד באתר. מצביעים יכולים להיכנס לעמוד Blockchain ומנהלים יכולים להיכנס לעמוד Vote (אנחנו רצינו שהרשאות מנהל יהיו לעמוד blockchain ומה שבתוכו בלבד ומצביעים יוכלו להיכנס לעמוד vote בלבד). הבעיה נוצרה מכך שלא ידענו שהרשאות משתמשים, authentication, יש לעשות מתחילת תהליך בניית הפרויקט ולהתקדם איתו תוך כדי עבודה. כאשר ניגשנו לנושא בסוף בנית הפרויקט נתקלנו בבעיות רבות שלא אפשרו לנו להתקדם בצורה טובה עם הרשאות משתמש.

Send Vote –

לאחר לחיצה על send vote לא מתנתקים מיד מהמערכת, הפתרון היה לעבור ישירות לעמוד הראשי ומשם ללחוץ על Logout. הבעיה לא נפתרה מכיוון ועדין ניתן לחזור לעמוד vote ולהזין הצבעה נוספת באתר. בנוסף, במידה ולא לוחצים על sent vote ניתן לחזור לעמוד vote להזין הצבעה נוספת והצבעה זו תתווסף לכרטיס ההצבעה ותהיה כמו הצבעה כפולה - ההצבעה לא תיפסל אבל הפתרון היה שהספירה תהיה על האלמנט הצבעה הראשון שמופיע בכרטיס.

הצבעה מרובה דרך פורטים שונים –

ההצבעה עוברת לפורטים אחרים לעמוד Vote Submitted ומחכה ללחיצת Submit. במציאות נרצה שבעת ההצבעה בקלפי הממוחשב ההצבעה לא תעבור לפורט (קלפי) אחר.

ספירת קולות –

ספירת קולות לא מתעדכנת באופן דינאמי אלא מתבצעת כל פעם מחדש ברגע לחיצת count votes. הרעיון הראשוני היה עדכון מידי ללא צורך בספירה מחדש, מה שקורה בפועל - מעבר מחדש על כל הבלוקים בבלוקצ'יין ובדיקת נתונים בכרטיס ההצבעה).

פתרון הפרויקט, תהליכים, אלגוריתמים ומה שביניהם

בעלי עניין:

- בעלי עניין ישיר-
  - אזרחי מדינת ישראל (אופציה למשאל עם) - ישתמשו במערכת להצבעה.
  - בחירות לארגון עובדים.
  - עיריות מוניציפליות וישובים.
  - פריימריז לחלק מהמפלגות בישראל.
  - בחירות לנשיאות ישראל.
  - בחירות לראשות הממשלה.
  - צוות מערכות מידע פנימי- אחראי להתקנה, תפעול ותחזוקת המערכת.
- בעלי עניין עקיף-
  - צוות בקרת איכות ופיקוח חיצוני על ההצבעות- מעקב ופיקוח על ההצבעות על פי החוק.
  - ספקי שרתים פנימיים לאחסון הנתונים- מספקת שירותים לשמירת נתונים וגיבוי המערכת של החברה.
  - צוות סייבר להגנת המערכת מפני תקיפות.
- השפעות שליליות-
  - תלות בטכנולוגיה.
  - הדרישה לעובדי קלפי תפחת.
  - מערכת בחירות מתחרה- תחרות בשוק.
  - הגברת שיח ציבורי בטענה לאי דיוקים וזיופים.

דרישות פונקציונליות:

- ❖ עמוד התחברות-
  - בעת כניסה למערכת האזרח יצטרך להכניס את פרטי תעודת הזהות שלו.
  - עמוד כולל:
    - קלט משתמש:
      - שם מלא.
      - תעודת זהות- 9 ספרות.
      - סיסמה.
      - צריך להכיל ספרות בלבד.
    - כפתור התחברות:
      - בעת לחיצה על כפתור ההתחברות תבוצע בדיקה בצד הלקוח שאכן הפרטים מולאו כראוי, במידה ולא נציג הודעה מתאימה: "הפרטים אינם נכונים".
      - במידה והקלט עבר בהצלחה את הבדיקות המשתמש יעבור לעמוד הראשי.
    - קלט הסיסמה:
      - תכיל עד 20 תווים.
      - הסיסמה תהיה ייחודית לכל משתמש, תכיל תווים שמתוכם יהיו לפחות תו אחד מיוחד, אות גדולה באנגלית, אות קטנה באנגלית.
      - בשלב מאוחר יותר הסיסמה תשלח באמצעות אסמס
- ❖ תפריט ראשי-
  - יופיעו הכפתורים הבאים:
    - תפריט אודות.
    - כפתור מעבר להצבעה.
    - כפתור "LOG OUT":
      - בעת לחיצה על כפתור ה"LOG OUT" המשתמש יצא מהמערכת.
- ❖ תפריט מעבר להצבעה:
  - בעת לחיצה על הכפתור "מעבר להצבעה" נעבור לחלון חדש שכולל:
    - "פתקי" המפלגות:
      - ספרות זיהוי.
      - שם המפלגה.
      - כפתור "Submit":
        - בעת לחיצה על "Submit" ההצבעה תקפוץ חלונית "Success" עם כפתור אישור.
        - לאחר לחיצת הכפתור האישור, המשתמש יעבור לתפריט "Vote Submitted".

- ❖ תפריט "Vote Submitted":
  - החלון יכלול:
    - תצוגה של בחירת המשתמש.
    - כפתור "SEND VOTE":
  - בעת לחיצה על "SEND VOTE" תקפוץ חלונית "Success" עם כפתור אישור.
  - לאחר לחיצה על כפתור האישור, המשתמש יחזור לתפריט האודות לבצע התנתקות.

דרישות לא פונקציונליות:

- ❖ Performances Requirements:
  - התחברות בו זמנית של מצביעים בהתאם למספר עמדות הצבעה.
  - מערכת המעבירה מידע בזמן אמת ובסיום תפיק דו"ח תוצאות אמת לא מאוחר משעה מסיום ההצבעה.
  - המערכת תהיה מהירה ותגיב למשתמש בזמן אמת ללא המתנה ארוכה.
  - קריסת המערכת לא תעלה על 2 פעמים ביום הבחירות.

Quality Attributes

- Reliability Requirements
  - סיסמת כל אזרח תהיה מוצפנת ותשלח לו עם פנקס הבוחר.
  - זמן התאוששות מקריסת מערכת, במידה ותהיה לא יעלה על 1 דקה.
  - זמינות המערכת תהיה בהתאם לשעות ההצבעה (7:00-22:00 לרוב).

- Usability Requirements
  - המערכת תהיה נגישה בשפה האנגלית.
  - בשלב מאוחר יותר המערכת תהיה נגישה בשפות: עברית, ערבית ורוסית.
  - בשלב מאוחר יותר המערכת תהיה מונגשת לבעלי מוגבלויות: הגדלת והקטנת פונטים, צבעים וכו'.
  - המערכת תהיה ידידותית למשתמש ותהיה קלה לתפעול.

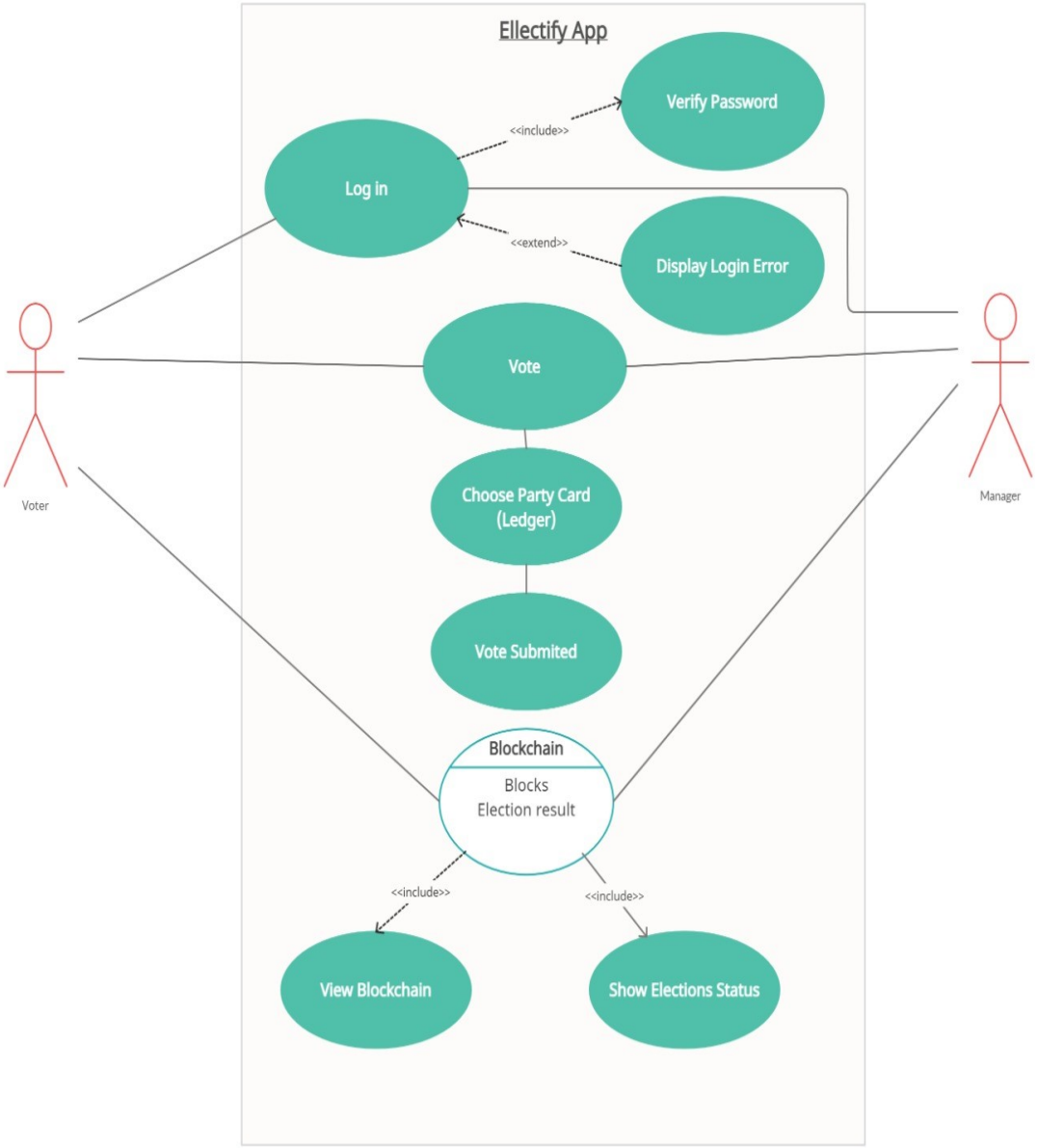
- Security Requirements
  - פרטיו האישיים של האזרח יהיו מוצפנים.
  - בחירתו של האזרח תהיה מוצפנת ומאובטחת.
  - קבלת סיסמה רנדומלית וחזקה שעומדת בתנאי אבטחה מחמירים.

- Maintainability Requirements
  - שמירת נתונים לאחר כל הצבעה.
  - גיבוי לשרת פנימי מאובטח לאחר כל הצבעה וכנ"ל בסיום הפקת הדו"ח.
  - המערכת תתנתק מהמשתמש לאחר הצבעה או אי שימוש לאחר כ-5 דק'.
  - בעת קריסת שרתי המערכת, כל הנתונים ישמרו ויגובו והמערכת תנתר לשרתי החירום על מנת למזער את שיבוש תהליך הבחירות.

- Supportability Requirements
  - המערכת תתמוך במחשבים ייעודים מטעם המדינה.
  - תרוץ על מערכת הפעלה Windows 8 ומעלה.
  - תרוץ על מערכת הפעלה Linux.

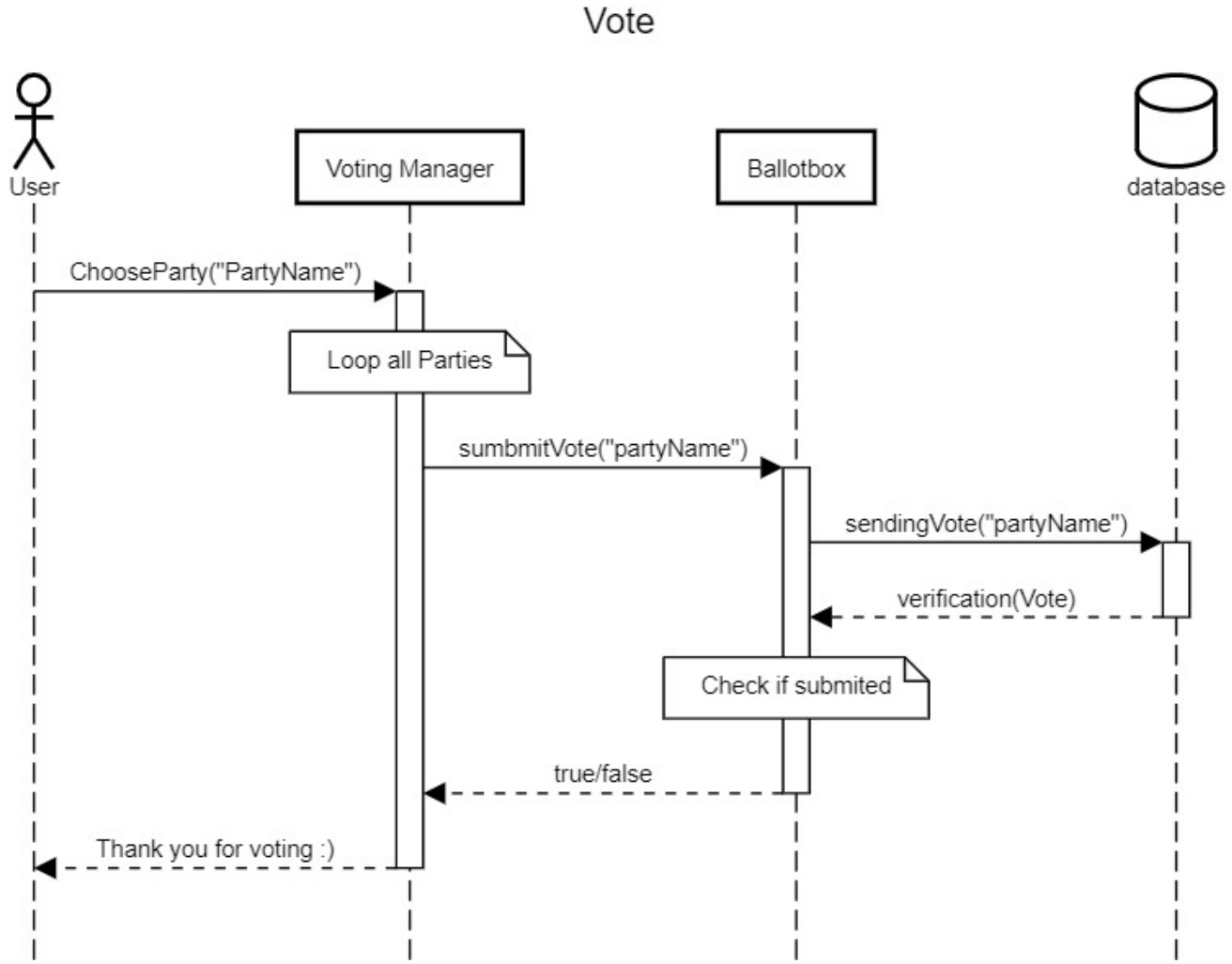
ארכיטקטורת UML -

Use Case -



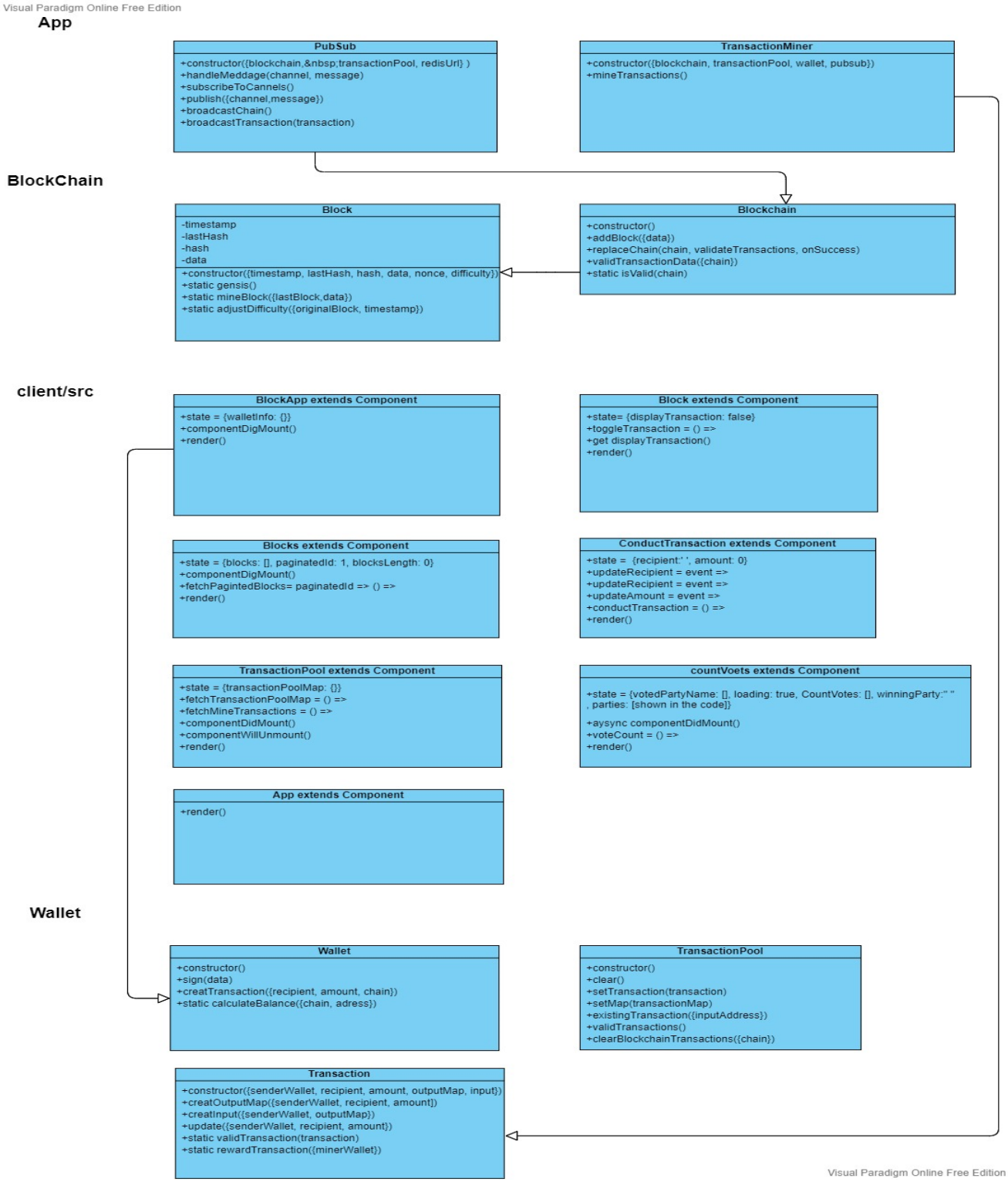
"איור 1, מתאר את רכיבי האפליקציה"

– Sequence Diagram



"איור 2, מתאר את שלבי ההצבעה"

- Class Diagram



"איור 3, ארכיטקטורת Classes. איור המראה את התלות והקשר בין כלל ה-Classes בפרויקט."

1. App-

- pubsub.js-
  - class PubSub- עובד בעזרת ערוצים בהם משתמשים יכולים להאזין להודעות ולמפצים לשדר את ההודעות שלהם.
    1. constructor ({blockchain, transactionPool, redisUrl})- יכיל 2 שדות לוקליים למשתנה מסוג Pubsub, שדה אחד יקרא publisher ולשני subscriber, נשים את שני השדות האלה תחת אותה מחלקה בשביל לאפשר למשתנה מסוג Pubsub להיות גם publisher וגם subscriber.
    2. handleMessage (channel, message)-פונקציית עזר שמתריעה על הודעה שהתקבלה, באיזה ערוץ היא התקבלה ומה תוכן ההודעה ובנוסף מחליפה שרשרת כאשר מתקבלת הודעה בערוץ הבלוקצ'יין, בנוסף מקבלת מטפלת בהודעות על עסקאות שאמורות להגיע.
    3. subscribeToChannels () - פונקציה שעוברת על כל ערוץ במפת הערוצים וקוראת ל- subscribe בכל ערוץ, דואג למנוי אוטומטי עבור ערוץ הבדיקה גם לערוצי הבלוקצ'יין וגם לערוצי העסקאות.
    4. publish ({channel, message})- עושה unsubscribe לערוץ ייעודי, לאחר מכן מפרסמים לו הודעה ואז נרשמים מחדש לערוץ הייעודי, מונע מpublishers לשלוח הודעות לא עקביות לאותו subscriber מקומי.
    5. broadcastChain () - פונקציה שמאפשרת לבלוקצ'יין לשדר את השרשרת שלה.
    6. broadcastTransaction(transaction)- משדר את העסקאות שבוצעו באותו הרגע.
- transaction-miner.js-
  - class TransactionMiner-מחזיק פונקציית עסקאות "מכרה" הקוראת לכל הפונקציות הדרושות להוספת בלוק של נתוני משיכת המידע של העסקאות לבלוקצ'יין.
    1. constructor ({blockchain, transactionPool, wallet, pubsub})- מכיל את המשתנים הנדרשים בשביל של-mineTransactions תהיה גישה אליהם.
    2. mineTransactions () - מקבל עסקאות חוקיות מה-transactionPool

2. BlockChain-

- block.js-
  - Class Block– מבנה כללי של הבלוקצ'יין, מכיל בתוכו את המשתנים: timestamp, lastHash, hash, data, nonce, difficulty.
    nonce ו- difficultyנוצרו בשביל לבדוק באיזו מהירות יכולים להבנות בלוקים חדשים במערכת.
    1. constructor ({timestamp, lastHash, hash, data, nonce, difficulty})- מאפשר לקבל ערכים בודדים עבור הבלוק
    2. static genesis- בלוק דמה (dummy block) של שרשרת הבלוקצ'יין.
    3. static mineBlock ({lastBlock, data})-עוזר למצוא Hash ולידי להוספת בלוק חדש.

4. static adjustDifficulty ({originalBlock, timestamp})- מחשב את ה-mine rate של כמה זמן לקח לכל בלוק להיחצב בשביל להגדיר את ה-difficulty

- <u>index.js-</u>
  - class Blockchain- אוסף את כל הבלוקים ומסדר אותם במערך שרשרת
    1. constructor () - נותן מערך שרשרת לכל מופע של בלוקצ'יין זה
    2. addBlock ({data})- יוצר את הקישוריות בין הבלוק הנוכחי לבלוק החדש שמתווסף לשרשרת
    3. replaceChain (chain, validateTransactions, onSuccess) - בודק האם בוצעה החלפה בין השרשרת הנוכחית לבין השרשרת שצריכה להתווסף למערך.
    4. validTransactionData ({chain})- מגדיר את החוקים של ההצפנה ברגע שצריכה להתבצע עסקה בין הבלוקים
    5. static isValidChain(chain) - בודקת עבור שרשרת שאמורה להצטרף לבלוק האם היא הצטרפה בהתאם לחוקים שהגדרנו ובעלת אותם מאפיינים של השרשרת הקיימת, תחזיר ערך False/True בהתאם

3. client/src -

- <u>BlockApp.js-</u>
  - class BlockApp extends Component- התצוגה הראשית של עמוד הבלוקצ'יין באפליקציה.
    1. state = {walletInfo: {}} - מקבלת בקשות מאובייקטים של ה-wallet הנמצאים ב-back end
    2. componentDidMount () - "lifecycle methods" המופעל מיד לאחר טעינת משתנה לתוך המסמך הראשי, ניעזר בזה בשביל למנוע חסימה של הופעת המצגת ב-render
    3. render () - מתודה שמחזירה את ה- JSX לפרטים כדי להסביר כיצד תוצג מחלקת הרכיבים של האפליקציה.

- <u>Blocks.js-</u>
  - class Blocks extends Component- התצוגה של הבלוקים, יבצע בקשה ב-blocks שב-backend שתחזיר מערך של בלוקים.
    1. state = {blocks: [], paginatedId: 1, blocksLength: 0} - מחזירה מערך ריק של בלוקים, עוקב אחרי ה-paginatedId של הרכיב בשביל שנוכל לקבוע איזה עמוד של בלוקים לייבא, ומכיל את אורך הבלוקים בשביל שהמשתמש יוכל לעשות קריאה חדשה לנקודת סף של עמוד הבלוק.
    2. componentDidMount () - אוסף את הנתונים המייצגים את הבלוקים של הבלוקצ'יין הנמצאים ב-back end.
    3. fetchPaginatedBlocks = paginatedId => () => - מייבא את הבלוקים עם ה paginatedId המבוקש.
    4. render () - מחזיר את המבנה הכללי של הבלוקים.

- Block.js-
  - class Block extends Component -מעבד את כל הבלוק, יוצר את המבנה הממשי לכל בלוק בנפרד.
  1. {displayTransaction: false} = state- עוזר לנו לעקוב אחרי תצוגת דגל העסקאות שכברירת מחדל הולך להיות False.
  2. <= () = toggleTransaction- מחליפה את מצב טרנסאקציית התצוגה לכל מי שהמצב הנוכחי שלו מנוגד
  3. () get displayTransaction- "computed properties" שמחזיר קומפוננט עסקה מלא לכל עסקה הנמצאת בשדה ה-data.
  4. () render-מקבל את המידע על הבלוקים דרך props.

- ConductTransaction.js-
  - class ConductTransaction extends Component- מאפשר למשתמש לשלוח עסקה מה-front בעזרת בקשת פוסט עם פרטי ה-recipient וה-amount ל-backend
  1. {recipient: ' ', amount: 0} = state- עוזר לעקוב אחרי ה-recipient, ה-amount וה-addresses שהמשתמש מקליד.
  2. <= (title, number) = chooseParty- בוחרת את המפלגה.
  3. <= event = updateRecipient- מעדכן את ה-recipient שב-state לפי מה שהמשתמש הקליד ב-FormControl.
  4. <= event = updateAmount- מעדכן את ה-amount שב-state לפי מה שהמשתמש הקליד ב-FormControl.
  5. <= () = conductTransaction- מחברת את בקשת ה-API לפרסום המידע של העסקה שהוכנסה על ידי המשתמש ב-backend.
  6. () render- מחזיר משתנה עם שם המחלקה.

- TransactionPool.js
  - class TransactionPool extends Component-מציג את העסקאות הנמצאות כרגע במאגר.
  1. {transactionPoolMap: {}} = state- מייצג את ערך החזרה שהופק של ה-transactionPoolMap.
  2. <= () = fetchTransactionPoolMap-מביא את ה-transactionPoolMap ומגדיר את התוצאה ב-state.
  3. <= () = fetchMineTransactions- מייבא שורת עסקאות שהולכות לבצע בקשה לכרות (mine) את העסקאות ב-backend.
  4. () componentDidMount- מאפשר לנתונים להיות זמינים במיידית בשביל שנוכל לייבא את העסקה מ-data ברגע שהם נטענים ל-HTML.
  5. () componentWillUnmount- מאפשר לנו להריץ את הקוד ברגע שהקומפוננט עוזב את ה-HTML ונמחק מהקובץ.
  6. () render-מחזיר משתנה ששם המחלקה שלו זה "TransactionPool".

- CountVotes.js
  - Class CountVotes extends Component- סופר את הקולות לאחר שהמשתמש בוחר את המפלגה שהוא רוצה.
    1. State = {votedPartyName: [], loading true, CountVotes: [], winningParty: " ", parties: [shown in the code]}- עוזר לעקוב אחרי המפלגות שנבחרו, מונה הקולות, מציג איזו מפלגה ניצחה ואיזה מפלגות יש.
    2. aysync componentDidMount () - מחכה לנתונים שיגיעו מה-fetch.
    3. voteCount = () => -מונה הסופר את הקולות שהתקבלו עבור כל מפלגה.
    4. Render () -מחזיר את המפלגה שניצחה ואת ספירת הקולות עבור כל מפלגה.

- App.js
  - Class App extends Component-מנהל את האפליקציה.
    1. Render () - מחזיר משתנה ששם המחלקה שלו זה "App" ומנתב את הקומפוננט לנתיבים הנכונים באפליקציה.

4. Wallet-

- index.js-
  - class Wallet- עוזר למשתמשים לתקשר אחד עם השני בחלק של ההצפנה בעזרת Keypair המכיל 2 סוגי מפתחות: Public key ו-Privet key .
    1. constructor () - מכיל משתנה של מאזן התחלתי וצמד מפתחות לכל ארנק.
    2. sign(data)- לוקחת ארגומנט נתונים ומחתימה אותו לתוך המפתח של אותה מתודת sign, לפני שאנחנו חותמים את המידע נבצע עליו hash וכך נבצע hash אופטימלי בכל פעם שנחתום על המידע.
    3. createTransaction ({recipient, amount, chain})-מאפשרת לארנק ליצור עסקאות משלו.
    4. static calculateBalance ({chain, address})-מחשב את המאזן בארנק לאחר ההעברה האחרונה שהתבצעה אצלו.

- transaction-pool.js-
  - class TransactionPool – אוסף עסקאות מתורמים שונים לאורך זמן ברשת הבלוקצ'יין.
    1. constructor () - יוצר מפת עסקאות
    2. clear () - מאפשרת ל-transactionPool להתנקות מעסקאות.
    3. setTransaction(transaction)- מגדיר את העסקה למפה בהתאם למספר מזהה
    4. setMap(transactionMap)- מגדירה את מפת העסקאות הלוקלית למפת העסקאות הנכנסת.
    5. existingTransaction ({inputAddress})- מחזיר את העסקה הקיימת עבור כתובת הקלט במידה ואם היא מאוחסנת ב-transactionMap.

6. () validTransactions-משיגה את כל העסקאות החוקיות העוברות בtransactionMap ומחזירה מערך של עסקאות חוקיות שעברו בדיקה ב-validTransaction שב-transaction.js.
7. ({chain}) clearBlockchainTransactions- מנקה את הבלוקים המכילים ערך בתוך ה- transaction ID.

- <u>transaction.js-</u>
  - class Transaction- מתעד את העסקות שנעשו בין שני בעלי ארנקים.

1. ({senderWallet, recipient, amount, outputMap, input}) constructor- מכיל אובייקט שנכנס.
2. ({senderWallet, recipient, amount}) createOutputMap-מכיל מפתח שקובע שה-recipient מקבל את ה-amount ובנוסף שה-publicKey מקבל את המאזן שיש בארנק.
3. ({senderWallet, outputMap}) createInput- פונקציית עזר המחזירה אובייקט הדומה ל- input field.
4. ({senderWallet, recipient, amount}) update-מוסיפה סכום חדש לrecipient חדש במפת הפלט של העסקאות הקיימות
5. (transaction)static validTransaction- בודקת האם העסקה חוקית או לא חוקית.
6. ({minerWallet}) static rewardTransaction- יוצרת עסקה עם הארנק של ה-miner ומשייכת את ה-reward לארנק שלו.

תשתיות וארכיטקטורות תוכנה:

לשם פיתוח השתמשנו בפלטפורמות, שיטות ושפות תוכנה הבאות:

- JavaScript.
- Node.js.
- HTML.
- CSS.
- React.
- Redis-

ניעזר בו לקבלה ושליחת מידע דרך ה-API, מספק מבני נתונים כגון: מחרוזות, bitmaps, hash ועוד.
מתאים למצבים הדורשים אחזור והעברת נתונים ללקוחות במהירות האפשרית.

- hexToBinary-

ספריה שיכולה להמיר מחרוזת מספרים בבסיס הקסדצימלי למחרוזת מספרים בבסיס בינארי בעלות אותו ערך.

- cryptoHash-

ניעזר ב- SHA 256 (Secure Hash Algorithm) מכיוון שזה בעל יתרונות כאשר מגיעים לחלק של הקריפטוגרפיה והאבטחה של הבלוקים, הוא מפיק ערך ייחודי לפלט ייחודי והוא פונקציה חד כיוונית ועל כך קשה לפרוץ אותו.

- ec-elliptic curve-

ניעזר ב-EC('spec256k1') המתייחס לפרמטרים של העקומה האליפטית המשמשת בקריפטוגרפיה של המפתח הציבורי של הביטקוין. ומוגדרת בסטנדרטים לקריפטוגרפיה יעילה.

- uuid-

ניעזר ב- ('uuid/v1'), UUID מהגרסה ה-1 של הוא אלגוריתם מזהה ייחודי אוניברסלי שנוצר באמצעות חותמת זמן וכתובת ה-MAC של המחשב עליו הוא נוצר.

- Postman API tool-

ניעזר בו לקבלת ושליחת מידע דרך ה-API.
כלי פיתוח שנועד לסייע בבניה, בדיקה, ושינוי ממשקי API.
כמעט כל פונקציונאליות שיכולה להיות נחוצה מוצפנת בכלי זה, בעל יכולות לבצע סוגים של בקשות HTTP: PATCH, PUT, POST,GET , שמירת סביבות לשימוש מאוחר יותר, המרת ה-API לקוד עבור שפות שונות.

- Express-

מסגרת מינימלית וגמישה של יישומי אינטרנט המספקת שכבה דקה של תכונות יישומי אינטרנט בסיסית מבלי לטשטש את התכונות של ה-Node.js.

- History-

ספריית history מאפשרת לנהל את היסטוריית הפעולות ב-JavaScript.
אובייקט history מפשט את ההבדלים בסביבות שונות ומספק ממש API מינימלי המאפשר לנהל את מחסנית ההיסטוריה, לנווט ולהמשיך את המצב בין הפעולות.

- Parcel-

ניעזר ב-Parcel לפתיחת השרת והאתר באותו פורט: localhost 3000.

Parcel הוא אוסף יישומי אינטרנט המציע ביצועים מהירים תוך שימוש בעיבוד רב ליבות הדורש קונפיגורציה אפסית.

- Jest-
מסגרת בדיקות ב-JavaScript שנועד להבטיח את נכונותו של בסיס הקוד, מאפשר לכתוב בדיקות עם API הנותנות תוצאות מהירות.
- app.get-
פונקציה זו מנתבת את בקשות ה-HTTP GET לנתיב שצוין עם פונקציות החזרה.
- app.post-
מנתבת את בקשות ה-HTTP POST לנתיב שצוין עם פונקציות החזרה.

בספריית React נעזרנו בכלים:

- Component-
הופכים את בניית ה-UI לקלה בהרבה, ניתן לעבוד עם כל קומפוננט באופן עצמאי ולמזג את כולם לקומפוננט אב שיהווה את ממש ה-UI הסופי.
- React-DOM-
מספקת שיטות ספציפיות ל-DOM שניתן להשתמש בהן ברמה העליונה של האפליקציה וגם כאפשרות לצאת ממודל ה-React במקרה הצורך.
- React-scripts-
קבוצת סקריפטים מחבילת ההתחלה create-react-app.
מגדיר את סביבת הפיתוח ומפעיל שאת ובנוסף מבצע טעינה מחדש של המודלים החמים.
- NavLink מ- 'react-router-dom'-
גרסה מיוחדת של <Link> שתוסיף תכונות עיצוב לאלמנט המעובד (rendered) כאשר הוא תואם את כתובת האתר הנוכחית.
- Button מ-'react-bootstrap'-
מאפשר למשתמש להוסיף כפתורי HTML סטנדרטים לפרויקט.
- FormGroup מ-'react-bootstrap'-
הרכיב עוטף form control עם ריווח נכון יחד עם תמיכה בתווית, טקסט עזרה ומצב אימות.
- FormControl מ-'react-bootstrap'-
מעניק form control עם עיצוב Bootstrap.
- withRouter מ-'react-router-dom'-
מעביר את המסלול הקרוב ביותר, המיקום הנוכחי ואביזרי ההיסטוריה לקומפוננט העטוף בכל פעם שהוא עובר עיבוד.
- Hooks-
מאפשרים שימוש ב-state ובעוד תכונות של React ללא צורך בכתיבת Class, בנוסף הם גם backward-compatible.
- useeffect-
מודיע ל-React שהקומפוננט צריך לבצע פעולה לאחר פעולת ה-render.
- usestate-
מחסל את תופעות הלוואי של שימוש בקומפוננט מבוסס CLASS.

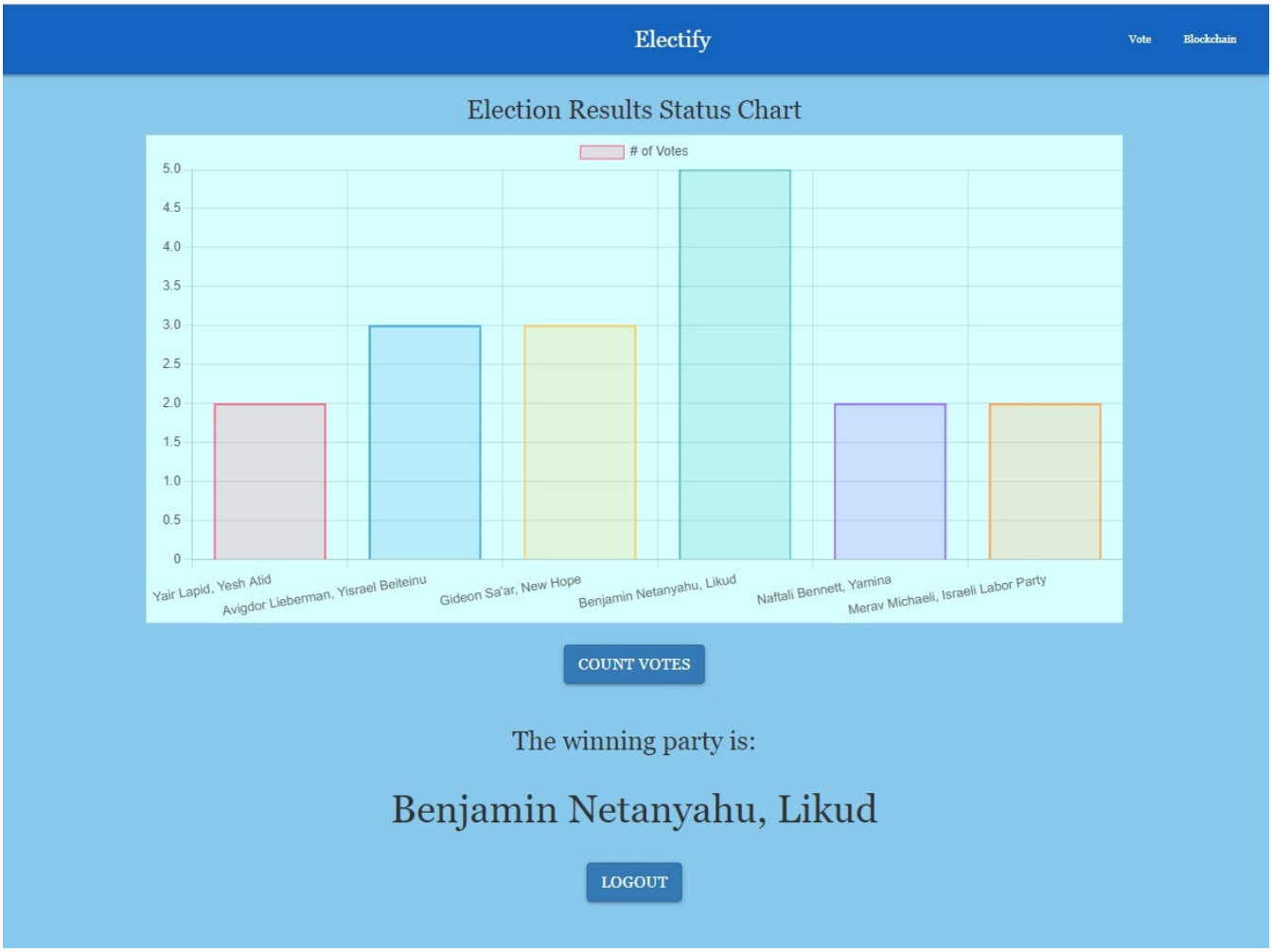
תוצרי הפרויקט

פירוט כלל תוצרי הפרויקט –

התוצרים שהתקבלו במהלך עבודה דו-סימסטריאלי:

- קלפי אלקטרוני חכם מבוסס חוזים חכמים ובלוקצ'יין המדמה את מערך הבחירות.

- מערכת בחירות מתקדמת וחדשנית מאובטחת בעלת יכולת למתן תוצאות בזמן אמת.



סיכום ומסקנות

התקשרות עם המנחה רועי זימון והרכבת הצוות החלה בחודש פברואר 21, שם לראשונה הוצג בפנינו אופי, מטרות וכיוון הפרויקט. הכיוון הכללי היה ברור לשני הצדדים והוא פרויקט בנושא חוזים חכמים ובלוקצ'יין אשר יאפשרו לנו לעשות טרנזקציות - העברות כספים, נכסים, מניות וכל סוג ערך אחר בצורה שקופה ללא קונפליקטים.

למען ההגינות, אנחנו כצוות קיבלנו חופש פעולה רחב ואוזן קשבת מהמנחה רועי זימון, מהסיבה שבתוכנית המקורית לפרויקט (טופס הצעת פרויקט) דרישת הפיתוח הייתה – פיתוח סוכנים חכמים באופן מבוזר.

כאמור, הרעיון הכללי מאוד דיבר אלינו וזאת הסיבה שבחרנו לקחת את הפרויקט הנ"ל בשתי ידיים ולעשות אותו. אך משהו היה חסר לנו, כמו שציינו בספר פרויקט מוקדם יותר, התקופה בה "נפגשנו עם הפרויקט" הגיעה במהלך בחירות לכנסת ה-24 שלמעשה גררו את מדינת ישראל בפעם ה-4 לבחירות תוך פחות משנתיים, תוך ציפייה שגם בבחירות הנ"ל לא תהיה הכרעה ונמשיך עם הסחרור הפוליטי.

חשבנו לעצמנו במשך כמה ימים, על כמויות המשאבים והכספים העצומים שמושקעים בכל מערכת בחירות שכזו, בחישוב גס, ארבע מערכות בחירות עלו למשלם המיסים כ-10 מיליארד שקלים, סכום בלתי נתפס שיכל ללכת לחינוך, רפואה, קשישים, תשתיות, נכים, אוכלוסיות קשות יום וכו'.

לכן, באישור של המנחה רועי זימון קיבלנו החלטה, לעשות "Pivot" קל לרעיון המקורי וללכת לכיוון של חוזה חכם לבחירות חכמות מבוססות בלוקצ'יין וקריפטוגרפיה.
ניתן להגיד שליבת הפרויקט לא השתנתה וזהה לרעיון המקורי בטופס הצעת פרויקט אך הייעוד הסופי כן שונה.

הצלחנו לעמוד ברוב היעדים והמטרות שהצבנו לעצמנו לצורך הנגשת הרעיון ויישום שלו בעתיד. הדבר נעשה באמצעות עבודה קשה, עמידה בתוכנית העבודה ולוח הזמנים שהגדרנו. בעתיד נמשיך לפתח ולעבוד על הרעיון, תוך התייעצויות עם אנשי מפתח ומחקר לצורך בדיקת התכנות אמיתית, מתוך אמונה שרעיון כזה, זאת לא שאלה של איך, אלא שאלה של מתי רעיון כזה יבשיל ויהיה מוכן לשימוש הן מצידנו והן מצד הממשלה והציבור.

בסופו של דבר הגענו לתוצר בעל פוטנציאל למימוש עתידי למערכת בחירות לראשות הממשלה ולכל רשות מוניציפלית אחרת, אך בפרק "תכנון וביצוע הפרויקט", תחת סעיף 6.7.3, צצו מספר בעיות שנתקלנו בהן במהלך הפרויקט ולכן ויש עוד דרך להגשמת הרעיון ומימוש הפרויקט.

תכנית עבודה של הפרויקט –

נציג את תכנית עבודה מפורטת של הפרויקט, לוח זמנים וחלוקת תפקידים.

1) פגישת צוות ראשונה יחד עם המנחה רועי זימון – כולם (עד ה- 22.2).
2) הגדרת מסמך דרישות והכנת מסמך אפיון – כולם (עד ה- 3.3).
3) איסוף מידע וסקירת ספרות – כולם (עד ה- 31.3).
4) תכנון הממשק הרעיוני - כולם (עד ה- 30.4).
5) פיתוח אב טיפוס של המערכת - כולם (עד ה- 30.6)
6) ספר פרויקט, פוסטר, מצגת – כולם (עד ה-31.7)
7) סרטון – כולם (עד ה-28.8)
8) תיקון דרישות, פנשים ובדיקה של עמידה בדרישות – כולם (עד ה-3.9)
9) הגשת תוצר סופי – כולם (עד ה-5.9)

תאריכים אלו נכתבו כתאריכי יעד אך אין זה אומר שלא נעשו שינויים ושיפורים עד שבוע אחרון להגשת הפרויקט. בנוסף חברי הצוות מכירים מהיום הראשון ללימודים במכללה ולכן הייתה סינרגיה ושת"פ מעולים, כל תהליך בפרויקט הוסכם על כולם.

המלצות להמשך פיתוח –

1) בעת התחברות למערכת המשתמש יקבל הודעת SMS עם הקוד הסודי.
2) שימוש עתידי בתעודות זהות או דרכון ביומטרי וטביעת אצבע.
3) שימוש ב-BI וניתוח נתונים בזמן אמת (פילוח מגדרים, ערים וכו').
4) מתן הרשאות למשתמשים שונים באפליקציה: מנהלי מערכת, עובדי קלפי, שרתים ומצביעים.
5) הצבעה ביתית באמצעות הנגשת המערכת לטאבלטים וטלפונים חכמים.
6) שיפור ממש ועיצוב הפלטפורמה.

איורים וסימונים –

1) Use Case, מתאר את רכיבי האפליקציה, איור 1.
2) Sequence Diagram, מתאר את שלבי ההצבעה, איור 2.
3) Class Diagram ארכיטקטורת Classes. איור המראה את התלות והקשר בין כלל ה-Classes בפרויקט, איור 3.

ביבליוגרפיה

https://www.amazon.com/Cryptopia-Bitcoin-Blockchains-Future-Internet/dp/B08HSLGR29

https://www.investopedia.com/terms/b/blockchain.asp

https://www.euromoney.com/learning/blockchain-explained/what-is-blockchain

/https://www.cryptojungle.co.il/ledger-nano-s-2

https://he.wikipedia.org/wiki/%D7%A7%D7%A8%D7%99%D7%A4%D7%98%D7%95%D7%92%D7%A8%D7%A4%D7%99%D7%94

https://he.wikipedia.org/wiki/Globally_Unique_Identifier

/https://www.geeksforgeeks.org/introduction-postman-api-development

/https://www.geeksforgeeks.org/reactjs-useeffect-hook

https://reactjs.org/docs/hooks-intro.html

/https://expressjs.com

https://www.npmjs.com/package/history