****

**מחלקת הנדסת תוכנה**

שם הפרויקט: סוכן רשת אוטומטי

Project Name: Automatic Web Agent

דו"ח ביניים 1

|  |  |
| --- | --- |
| שם הסטודנט: | קובי הרשקוביץ |
| מספר תעודת זהות: | 025632654 |
| שם המנחה: | אלעד הוגן |
| תאריך ההגשה: | 8/9/09 |

**תוכן ענינים**

[מבוא 4](#_Toc229036130)

[מטרות, יעדים ומדדים 5](#_Toc229036131)

[סקירה ספרותית 6](#_Toc229036132)

[סקירת מצב קיים 7](#_Toc229036133)

[מסמך דרישות ראשוני ותיחום של המערכת 7](#_Toc229036134)

[גורמי הפרויקט 7](#_Toc229036135)

[תכלית הפרויקט 7](#_Toc229036136)

[לקוחות הפרויקט 7](#_Toc229036137)

[משתמשי הפרויקט 7](#_Toc229036138)

[אילוצי הפרויקט 7](#_Toc229036139)

[אילוצים ראשיים/קריטיים 7](#_Toc229036140)

[הגדרות שמות, קיצורים וראשי תיבות 7](#_Toc229036141)

[עובדות והנחות הקשורות לפרויקט 8](#_Toc229036142)

[דרישות פונקציונאליות 8](#_Toc229036143)

[תחום העבודה 8](#_Toc229036144)

[תחום המוצר 8](#_Toc229036145)

[דרישות מידע ופונקציונאליות 8](#_Toc229036146)

[דרישות לא פונקציונאליות 8](#_Toc229036147)

[דרישות עיצוב 8](#_Toc229036148)

[דרישות שימוש וממשק משתמש 8](#_Toc229036149)

[דרישות ביצועים 8](#_Toc229036150)

[דרישות הפעלה 8](#_Toc229036151)

[דרישות תחזוקה ותמיכה 9](#_Toc229036152)

[דרישות אבטחה 9](#_Toc229036153)

[דרישות תרבותיות ופוליטיות 9](#_Toc229036154)

[דרישות חוקיות 9](#_Toc229036155)

[נושאי הפרוייקט 9](#_Toc229036156)

[נושאים פתוחים 9](#_Toc229036157)

[פתרונות מדף 9](#_Toc229036158)

[בעיות חדשות 9](#_Toc229036159)

[משימות 9](#_Toc229036160)

[העברת המערכת לידי/לניהול הלקוח 9](#_Toc229036161)

[סיכונים 9](#_Toc229036162)

[עלויות 9](#_Toc229036163)

[תיעוד ולימוד משתמשים 10](#_Toc229036164)

[דרישות לדור הבא 10](#_Toc229036165)

[רעיונות לפתרון 10](#_Toc229036166)

[ניתוח פונקציונאלי ראשוני 10](#_Toc229036167)

[ניתוח חלופות מערכתי ראשוני 13](#_Toc229036168)

[תאור דרך הביצוע המתוכנן 13](#_Toc229036169)

[האמצעים/הכלים הנדרשים 13](#_Toc229036170)

[פערי ידע שעל הסטודנט להשלים 14](#_Toc229036171)

[תוצרי הפרויקט 14](#_Toc229036172)

[תכנית בדיקות ראשונית 15](#_Toc229036173)

[ניהול סיכונים 15](#_Toc229036174)

[תוכנית ניהול הפרויקט 16](#_Toc229036175)

[Work Breakdown Structure 16](#_Toc229036176)

[תוכנית עבודה של הפרויקט 17](#_Toc229036177)

[רשימת מקורות: 17](#_Toc229036178)

# תקציר

כיום השימוש באינטרנט נעשה רווח כמעט בכל בית ובאמצעותו ניתן לבצע משימות רבות ומגוונות החל מקבלת מידע בידורי וכלה בפעולות בנקאיות. במרבית מן הפעולות נדרש המשתמש להזדהות, לבצע מעקב אחר נתונים באתר, ולבצע פעולות על סמך ידע שלמד מאתרים אחרים. פעולות אלו במרבית הזמן זהות/דומות בין האתרים וכמו כן ביצוען עלול לדרוש תשומת לב וזמן רב מן המשתמש.

הפרוייקט המוצע הינו סוכן רשת אוטומטי הבא להחליף מטלות שגרתיות וגוזלות זמן של משתמש דפדפן ממוצע/מתקדם. באמצעות המערכת המוצעת יוכל המשתמש לתכנן תסריט של פעולות מסויימות באתרים מסויימים בהתאם למידע הנדגם מאתרים אחרים ולקבוע את זמן ביצוע התסריט.

לדוגמא ניתן לקבוע תסריט שבו נדגם כל 4 שעות אתר סקי שהמשתמש מחבב במיוחד (לדוגמא www.champery.com) , התסריט מכוון לזהות את השדה של גובה השלד באתר וברגע שגובה השלג מגיע ל-110cm התסריט שולף מאתר ה-"דקה ה-90" את מחירי חבילות הנופש לאתר ושולח דוח מפורט עם הנתונים באי-מייל למשתמש.

דוגמא נוספת יכולה להיות תסריט שיאריך את זמן השאלת ספר באתר הספריה של מכללת אפקה ,שאינו מצוייד כיום בסוכן אוטומטי להארכת השאלות, פעם בשלושה ימים.

יחודיות הפרוייקט היא שבניגוד לסוכנים אוטומטיים הנמצאים באתר מסויים ונשלטים מתוך האתר הפרוייקט הנ"ל מציע סוכן שיתוכנת בצד הלקוח ולא בצד השרת, דבר אשר יאפשר לו נוחויות ושליטה מירבית להשגת מטרותיו.

לחילופין במישור העסקי, חברה לממכר מוצרים באיטרנט יכולה לבצע מעקב אחר מוצרים מקבילים באתרי מתחרים ולעדכן את המחירים באתר שלה ע"פ התוצאות הנדגמות.

# סקירה ספרותית

1. סוכני רשת הם מערכות תוכנה מורכבות אשר פועלות ברשת האינטרנט העולמית (www), ברשת הארגונית או ברשתות מקומיות. סוכני רשת מתוכננים לבצע מגוון משימות החל מאגירה וניתוב של מידע וכלה בחיפוש קיטלוג וסינון מידע.

הספרות מתארת מסגרת לתכנון וניתוח של סוכני רשת ומערכות סוכנים בהתבסס על מודלים מתמטיים של הסביבה בה הם פועלים. יש 3 אבני בניין מרכזיות לבניית סוכני רשת:

* מודלים סטוקאסטיים של גרף הרשת אשר משקללים את הפילוג והקישוריות של דפי רשת ונותנים קוים מנחים מרכזיים בתכנון סוכנים, ע"י הפקת תכונות ייחודיות של הסביבה בה הסוכנים פועלים.

הרשת יכולה להיראות כגרף מכוון אשר הצמתים בו הם דפי HTML סטטיים והקשתות בו הם קישורים מאתר אחד למשנהו. מאמר [ ‎18] מציע ומנתח מודלי גרפיים אקראיים שנוצרו ע"י תצפיות נסיוניות ברשת. המודלים שונים מהמודלים המקובלים בכך ש קשתות שנבחרו בצורה בלתי תלויה לא משתקפות בסטטיסטיקה שנצפתה ברשת, כך שקשתות במודלים אלו הינם תלויות סטטיסטית ביניהן. כמו כן נוספים צמתים לגרף כתלות בזמן. דבר זה משקף את העובדה שהרשת משתנה עם הזמן.

* שיטות למידה סטטיסטית המאפשרות לסוכנים ללמוד על סביבתם ע"י הפקת מודלים סטוקאסטיים של דף רשת ומבנה קישורים מקומיים. מאמר [ ] מתאר את הרשת כעולם שבו המוני קישורים מחברים בין מסמכי ודפי אתרים כך שנוצר גרף בעל סיבוכיות גבוהה וללא תקדים וזהו גרף הרשת. מאמר זה מציג גישה חדשה ללימוד מודלים הסתברותיים שך הרשת אשר יכולים לשמש כחלק בתהליך חיזוי קישוריות ומידע של מסמכי רשת. השיטה המתוארת במאמר זה היא שיטה הסתברותית לצמצום מימדים (probabilistic dimension reduction) אשר גוזרת ומאחדת את שיטת הניתוח הסמנטי המאוחר (Latent Semantic Analysis) ואת אלגוריתם קליינברג (Hubs-and-Authorities) בסביבה סטטיסטית.

דבר זה אמור להיות הצעד הראשון לפיתוח תשתית סטטיסטית של טכנולוגיות מידע ברשת. למרות שמאמר זה לא מתמקד ביישום מסויים, מגוון אלגוריתמים הפועלים בסביבת רשת יכולים לנצל את היתרונות בטכניקות המוצגות למטרות שונות כגון מנועי חיפוש, "זוחלי רשת" (Web crawlers) ומערכות סוכני מידע.

* אלגוריתמים לתכנון אוטונומי וקבלת החלטות בסביבת רשת לאפשר לסוכנים להשיג מטרות ולהסתגל לסביבה משתנה [16]

ניקח לדוגמא בעיות קלאסיות בתחום בסיסי נתונים/איחזור מידע כגון חיפוש בטקסט, כריית מידע וסיווג. הקלט לבעיות מסוג זה הוא בדרך כלל אוסף של נתונים/מסמכים. הרשת עם המבנה הנוסף שלה כגרף, מאפשרת את השיפור של טכניקות קיימות עם טכניקות מעולם הגרפים. במאמר הודגם שיפור זה באמצעות פתרונות מבוססי גרפים לבעיות הבאות: חיפש נושא, מיספור נושא, סיווג וזחילה (crawling).

1. ככל שאפליקציות רשת נעשו מסובכות יותר, בדיקתן נעשתה יותר חשובה. יש מספר רב של טכניקות בדיקה הנגישות למתכנת, לדוגמא במהדורת אפריל 2005 של *MSDN*®*Magazine* יש תאור של מערכת בדיקה מבוססת Jscript שיכולה לבדוק אפליקציית רשת באמצעות ממשק המשתמש שלה על-ידי שימוש במודל האובייקטים של IE (DOM) טכניקה זו יעילה אך בעךת מספר מגבלות. במאמר זה מתוארת טכניקה בה משתמשים ב-DOTNET ליצירת בדיקת אפליקציית רשת כאשר הספריות המשמשות ברמה הנמוכה הן fmshtml.dll ו- shdocvw.dll שתפקידן לגשת ולשנות את ה-DOM.[20]

# מטרות העבודה

**מטרות:**

רבות מהפעולות הפיננסיות, הלימודיות והבידוריות נעשות באמצעות דפדפן ברשת האינטרנט:

* רישום לאתרים
* קניה/מכירה של מניות
* הזמנת כרטיסים
* הגשת מטלות
* בחירת קורסים

רבות ממטלות אלו הן מחזוריות, שגרתיות ועלולות להתיש את המשתמש ו/או לגרום לו לבצע שגיאות באופן הכנסת הנתונים.

מטרת הפרוייקט הנגזרת מבעיה זו היא בניית אפליקציה אשר תבצע מטלות שגרתיות וצפויות מראש בצורה אוטומטית במקום משתמש דפדפן אנושי

**יעדים:**

ניתן לפרוט מטרת על זו ל-3 יעדים:

* ניטור אתר בצורה מחזורית
* ביצוע פעולות אוטומטיות בדפדפן
* התראה למשתמש ע"פ בקשתו על אירוע באתר מסויים.

**מדדים:**

המדדים ע"פ תימדד הצלחת הפרוייקט הם:

* אחוז האתרים הנתמכים מתוך עשרת האתרים הראשונים בחיפוש בגוגל יעמוד על 80% בקטגוריות פיננסים, בידור, ספורט (דוגמא לחיפוש יכולה להיות המילה 'פיננסי')
* נוהל כניסה לאתר הספרייה במכללה והארכת השאלה אם קיימים ספרים מושאלים יסתיים בהצלחה.
* נוהל דגימת אתר לשינויים בתדירות של 6 פעמים בדקה יסתיים בהצלחה (ידווח בצורה נכונה על שינוי).
* משתמש חדש יוכל להגדיר תסריט אוטומטי של דגימת 2 אתרים ואינטראקציה ביניהם בתוך כשעה.

# סקירת מצב קיים

כיום כאשר משתמש רוצה לבצע מעקב/שינוי נתונים באתר מסויים פרושות לפניו מספר אפשרויות:

תמיכה של האתר עצמו בסוכן רשת אשר מיידע את המשתמש לגבי נתון ייחודי לאתר אליו הוא שייך. לדוגמא האתר ebay מציע סוכן שיציע במכרז מחיר עבור פריט מסויים עד תקרה שיקבע המשתמש.

מעקב מחזורי אישי של המשתמש אחר שינויים באתר.

שימוש בתוכנות אוטומציה יקרות כדוגמת QFT או RFT.

QTP(Quick Test Pro ) היא חבילת תוכנה מבית מרקורי שבה ניתן לבצע אוטומציה לאפליקציות חלונאיות ורשתיות כגון אוטומציה לכתיבת דואר אלקטרוני ושליחתו או חקירת דף רשת והכנסת נתונים אליו.

RFT (Rational Functional Tester) היא גם חבילת תוכנה לבדיקות אוטומטיות מבית IBM המאפשרת גם לבצע בדיקות על אלמנטים מסוג Flex וטרמינלים כגון VT102.

מחירם של כל אחת מחיבולות אלו מגיע לכמה אלפי דולרים עבור רשיון למשתמש אחד.

# ניתוח חלופות מערכתי

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| חלופה | קריטריון להשוואה | יתרון | חסרון | משקל (חשיבות החלופה יחסית לכלל הפרוייקט) |
| 1. הקלטת אלמנטים עבור בסיס הנתונים באמצעות WebControl | ייצור בסיס הנתונים עבור האלמנטים של האתרים . | קלות בבחירת האלמנטים (WYSIWYG) | סיבוכיות בתכנות ממשק המשתמש | 30% |
| 1. הכנסת אלמנטים לבסיס הנתונים באמצעות טופס מילוי | קלות בתכנות צורת קלט זו | יש צורך במומחיות מסויימת של המשתמש וביכולתו לנתח את הדף.(לדוגמא באמצעות view source) |
| 1. כתיבת תסריט באמצעות שפת תסריטים ומפרש | תסריט פעולה | גמישות בפעולות האפשריות לביצוע על אתרים | על המשתמש ללמוד שפת תסריט, כמו כן נוצרת סיבוכיות של יצירת מפרש (תכנות מורכב) | 50% |
| 1. ייצור התסריט באמצעות טופס מובנה ובקרים | קלות בייצור התסריט מבחינת המשתמש | מגוון פעולות מוגבל יחסית לחלופת התסריט החופשי. |
| 1. עבודה מול דפדפן מחוץ לאפליקציה | ממשק משתמש | זהו עולם מוכר למשתמש | קושי בהתממשקות לדפדפן חיצוני | 20% |
| 1. עבודה מול דפדפן המשובץ בתוך האפליקציה |  |  |

טבלה 1

**סיכום**: בכדי לעמוד בלוח הזמנים תוך כדי שמירה על איכות נבחרו הרכיבים הבאים מבין החלופות – 1,4,6.

חלופה 1 נבחרה מכיוון שכבר ערכתי ניסוי לבחירה אינטראקטיבית של אלמנט מדף והתרשמתי שיהיה יחסית פשוט למימוש בצורה זו.

חלופה 4 נבחרה מכיוון שהסיבוכיות ביצירת שפת תסריט ומפרש לה מהווה סיכון מבחינת הצלחת המשימה ועמידה בזמנים.

חלופה 6 נבחרה משום שבדפדפן מובנה יש שליטה רבה יותר כמו כן כך המשתמש יידע בוודאות מהם מרכיבי האפליקציה.

# ניתוח חלופות טכנולוגיות

כדי לממש את המערכת יש צורך במספר טכנולוגיות שונות:

* שפת התכנות בה ייכתב הקוד – הן מבחינת יעילות השפה הן מבחינת קלות הכתיבה בה והן מבחינת עקומת הלימוד.
* סביבת הפיתוח
* כיצד ובאיזה מבנה יישמרו התסריטים/אלמנטי האתר – סיבוכיות המימוש, יעילות הגישה לנתונים.
* ספריית הגישה למודל האובייקטים בדף (DOM Library)

בטבלה הבאה מרוכזים החלופות השונות , הציון הניתן לכל חלופה הוא מ-1 עד 10 כאשר 10 הוא הטוב ביותר.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| נושא לחלופה | חלופה 1 | | | | חלופה 2 | | | | חלופה 3 | | | |
| שם | יתרונות | חסרונות | ציון | שם | יתרונות | חסרונות | ציון | שם | יתרונות | חסרונות | ציון |
| שפת תכנות  משקל:25% | CPP | יעילה, רוב ספריות API כתובות בה | מסובכת יחסית | 7 | JAVA | פורטבילית, קל לכתוב בה | ספריה גרפית מגושמת, פחות יעילה | 6 | C# | המוכרת לי ביותר, כתיבה פשוטה, ספריות גרפיקה ותקשורת עשירות | נתמכת בעיקר בסביבת windows | 9 |
| סביבת עבודה  משקל:15% | MS Visual Studio | תעשייתית, ממשק משתמש נוח, אמינה | עלות יקרה | 9 | Eclipse | חינם, קהילה רחבה של משתמשים | מותאמת בעיקר ל-JAVA | 8 | CodeBlocks | חינם | סביבה דלה במשתמשים מותאמת בעיקר ל-CPP | 5 |
| ספריית DOM Library  משקל:40% | Native (mshtml.dll shdocvw.dll) | שליטה מלאה בכל מה שניתן לעשות ב-DOM | כתובה ב-CPP שפה יותר מסובכת עבורי, ללא שליטה בדיאלוגים | 8 | Watin | מבוססת C# קלה לשליטה ובעלת תמיכה בדיאלוגים | לא תומכת בכל האפשריות של ה-DOM | 8 | Selenium | מבוססת JAVA פשוטה, תמיכה נרחבת | לא קל לשלב אותה באפליקציה חלונאית | 6 |
| מבנה נתונים משקל: 20% | MS SQL | תעשייתי, מתממשק ל-C# מיידית | מסובך | 8 | SQLITE | פשוט, קל משקל, מוכר, חינמי | יש תמיכה רק בחלק מ-SQL סטנדרטי | 9 | קבצי XML | חינמי, תמיכה רחבה. | שמירה מקומית | 9 |

# הפתרון שנבחר

לאחר שיקולי כל הגורמים על פי שקלול הציונים והמשקלות בטבלה הנ"ל נעשתה הבחירה הבאה:

* שפת תכנות: C# - השפה המוכרת לי ביותר, יש בה שילוב של עוצמה בעיצוב מהיר של אפליקציות חלונאיות ביחד עם WebControl נוח לשליטה.
* סביבת עבודה: Visual Studio 2008 – סביבה זמינה במכללה.
* ספריית DOM Library: Watin – ספריית קוד פתוח המפשטת טיפול בדיאלוגים ואלמנטים ואשר תחסוך זמן תכנות רב , כך שישאר יותר זמן לעסוק בלוגיקת המערכת מאשר בפרטים טכניים.
* מבנה נתונים: בסיס הנתונים Sqlite בשילוב עם ה-Sqlite Dotnet Provider – בסיס נתונים אמין ופשוט ובעיקר חינמי.

למערכת יהיו שלושה חלקים עיקריים ממשק הקלטת השדות הנדגמים, מודול ייצור התסריטים ומודול התזמון. דרך מימושם מתוארת בנספח ה-SRD (נספח ב1).

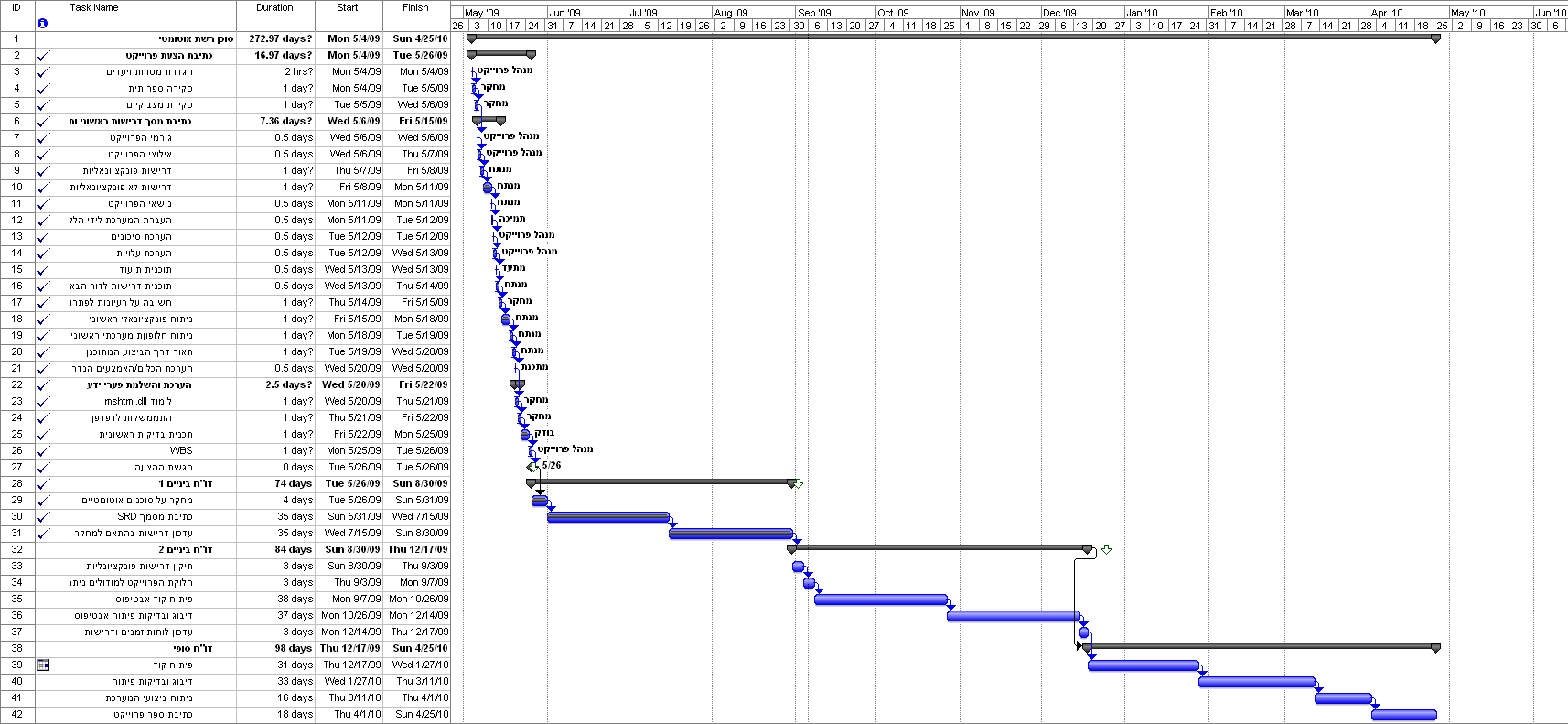
# תכנית בדיקות

סביבת הבדיקה הבסיסית תכיל 3 אתרי דמה בעלי מגוון פקדים, הבדיקות השונות יכללו נסיון להקליט פקדים ושדות מאתרים אלו, ליצור מספר תסריטים שיכללו את מגוון האפשרויות הניתנות לתסרוט, והפעלתם בתזמון מתוכנן מראש.

לתוכנית המפורטת ראה נספח ב2.

# תוכנית עבודה של הפרויקט

על פי התוכנית שהוגשה בהצעה (נספח 1)



# ניהול סיכונים

|  |  |  |  |
| --- | --- | --- | --- |
| סיכון | סיכוי | השפעה | דרך מניעה |
| חריגה מלוחות זמנים | 50% | אי הגעה לאבני דרך בפרוייקט | ניסיון להקדים ביצוע של משימות והקדשת זמן נוסף לפרוייקט |
| כשל חומרתי בסביבת הפיתוח/ריצה | 20% | עיכוב של עד כשבועיים | גיבוי כל איטרציה במערכת ניהול קוד |
| אי יכולת לזהות אלמנטים מיוחדים בדף | 50% | אי יכולת להשתמש באלמנטים אלו – ירידה בפונקציונאליות המובטחת | עדכון סעיף האילוצים בשלב מוקדם ככל האפשר |
| דפים דינמיים (Ajax, DHTML) | 80% | שיבוש יכולת זיהוי האלמנטים | התקנת מנגנוני המתנה לאלמנטים לא קיימים. |

# רשימת מקורות:

1. Jon Kleinberg. **Authoritative sources in a hyperlinked environment**. Proc. 9th ACM-SIAM Symposium on Discrete Algorithms, 1998. Extended version in Journal of the ACM 46(1999). Also appears as IBM Research Report RJ 10076, May 1997. [[pdf]](http://www.cs.cornell.edu/home/kleinber/auth.pdf)
2. Soumen Chakrabarti, Martin van den Berg, Byron Dom, **Focused Crawling: A New Approach to Topic-Specific Web Resource Discovery**, WWW8 [[html]](http://www8.org/w8-papers/5a-search-query/crawling/index.html)
3. Steve Lawrence and Lee Giles, **Searching the World Wide Web,** Science 1998 [[pdf]](http://www.neci.nj.nec.com/~lawrence/papers/search-science98/search-science98.pdf)
4. Jeffrey Dean, Monika R. Henzinger, **Finding Related Pages in the World Wide Web**, WWW8 [[html]](http://www8.org/w8-papers/4a-search-mining/finding/finding.html)
5. Oren Zamir and Oren Etzioni, **Grouper: A Dynamic Clustering Interface to Web Search Results**, WWW8, [[html]](http://www8.org/w8-papers/3a-search-query/dynamic/dynamic.html)
6. Rajan M. Lukose and Bernardo A. Huberman, **Surfing as a Real Option** [[abstract]](http://www.parc.xerox.com/spl/groups/dynamics/abstracts/InternetEcology/options.html)
7. Bernardo A. Huberman and Rajan M. Lukose, **Social Dilemmas and Internet Congestion** [[abstract]](http://www.parc.xerox.com/spl/groups/dynamics/abstracts/InternetEcology/storms.htm)
8. Bernardo A. Huberman, Peter L.T. Pirolli, James E. Pitkow, and Rajan M. Lukose, **Strong Regularities in World Wide Web Surfing**, Nature ??? [[abstact]](http://www.parc.xerox.com/spl/groups/dynamics/abstracts/InternetEcology/surfing.html)
9. William W. Cohen, Wei Fan, **Web-Collaborative Filtering: Recommending Music By Spidering the Web**, WWW9 [[html]](http://www9.org/w9cdrom/266/266.html)
10. Mike Perkowitz Oren Etzioni, **Towards Adaptive Web Sites: Conceptual Framework and Case Study**, WWW8 [[html]](http://www8.org/w8-papers/2b-customizing/towards/towards.html)
11. Marc Langheinrich, Atsuyoshi Nakamura, Naoki Abe, Tomonari Kamba, Yoshiyuki Koseki, **Unintrusive Customization Techniques for Web Advertising**, WWW8 [[html]](http://www8.org/w8-papers/2b-customizing/unintrusive/unintrusive.html)
12. Jim Hendler, **Is there an intelligent agent in your future?** Nature, Web matters, 11 March 1999. [[html]](http://helix.nature.com/webmatters/agents/agents.html)
13. Steve Kirsch, **The future of Internet search (keynote address)**, SIGIR 1999 [[pdf]](http://www.acm.org/pubs/articles/proceedings/ir/290941/p1-kirsch/p1-kirsch.pdf)
14. G. Pandurangan, P. Raghavan, and E. Upfal. Using PageRank to Characterize Web Structure, Proceedings of the 8th International Computing and Combinatorics Conference (COCOON), 2002.
15. [G. Pandurangan](http://www.cs.brown.edu/people/gopal) , [P. Raghavan](http://theory.stanford.edu/people/raghavan/index.html) , and [Eli Upfal](http://www.cs.brown.edu/people/eli), [**Building Low-Diameter P2P Networks**](http://www.cs.brown.edu/research/webagent/focs-2001.pdf). Proceedings of the 42th IEEE Symp. on Foundations of Computer Science. 2001.
16. S.R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and [Eli Upfal](http://www.cs.brown.edu/people/eli), [**The Web as a graph**](http://www.cs.brown.edu/research/webagent/pods-2000.pdf). Proceedings of the 19th ACM Symposium on Principles of Database Systems, pp 1-10, 2000.
17. Thomas Hofmann, [**Learning Probabilistic Models of the Web**](http://www.cs.brown.edu/research/webagent/Hofmann-SIGIR00.pdf), ACM SIGIR 2000
18. R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and [Eli Upfal](http://www.cs.brown.edu/people/eli), [**Stochastic models for the Web graph**](http://www.cs.brown.edu/research/webagent/focs-2000.pdf). Proceedings of the 41th IEEE Symp. on Foundations of Computer Science. 2000.
19. <http://en.wikipedia.org/wiki/Trident_%28layout_engine%29> , downloaded 1/9/2009
20. <http://msdn.microsoft.com/en-us/magazine/cc163723.aspx> , downloaded 1/9/2009

**נספח ב1 - (Software Requirements Document (SRD**

**(מבוסס על IEEE Std 830-1998)**

**1 Introduction**

**1.1 Purpose.**

The purpose of this document is to describe the requirement needed for the Automatic Web Agent System system.

The document is divided to sections, where as each section will describe the requirements from a single module.

**1.2 Scope of the software**

Nowadays web use has become common to practically every household. The web enables to perform many diverse tasks, ranging from acquiring entertainment data (movies, songs, gossip etc…) to financial transactions.

Most of the tasks a user might perform, require the user to authenticate his identity, track data on the website and perform actions based on knowledge acquired on other websites. These actions are very similar in most of the tasks regardless of their nature, all of which require the attention of the user and might be extremely time consuming.

The Automatic Web Agent application project was devised to relieve the average/advanced user from these routine and time consuming tasks.

The Automatic Web Agent will enable the user to compose a script of actions to be taken on specific websites according to data sampled on other sites and conditions that the users will set, and determine the scheduling of the script execution.

For example, there can be a script that will sample the users favorite sky website every 4 hours and will monitor the snow depth of the sky track. Once a preset depth is detected the script will extract the prices of plane tickets to the sky track and email them to the user.

Another example can be a script that will extend the lending period of a book in the Afeka college library, which is not equipped today with a built-in agent, once every 3 days.

Alternatively, an example from the business sector could be a for a company which sells products via the web, a script can track the competitors web sites for price changes and update the prices of the products accordingly.

**1.3 Definitions, acronyms and abbreviations.**

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Table

**1.4 References**.

1. Jon Kleinberg. **Authoritative sources in a hyperlinked environment**. Proc. 9th ACM-SIAM Symposium on Discrete Algorithms, 1998. Extended version in Journal of the ACM 46(1999). Also appears as IBM Research Report RJ 10076, May 1997. [[pdf]](http://www.cs.cornell.edu/home/kleinber/auth.pdf)
2. Soumen Chakrabarti, Martin van den Berg, Byron Dom, **Focused Crawling: A New Approach to Topic-Specific Web Resource Discovery**, WWW8 [[html]](http://www8.org/w8-papers/5a-search-query/crawling/index.html)
3. Steve Lawrence and Lee Giles, **Searching the World Wide Web,** Science 1998 [[pdf]](http://www.neci.nj.nec.com/~lawrence/papers/search-science98/search-science98.pdf)
4. Jeffrey Dean, Monika R. Henzinger, **Finding Related Pages in the World Wide Web**, WWW8 [[html]](http://www8.org/w8-papers/4a-search-mining/finding/finding.html)
5. Oren Zamir and Oren Etzioni, **Grouper: A Dynamic Clustering Interface to Web Search Results**, WWW8, [[html]](http://www8.org/w8-papers/3a-search-query/dynamic/dynamic.html)
6. Rajan M. Lukose and Bernardo A. Huberman, **Surfing as a Real Option** [[abstract]](http://www.parc.xerox.com/spl/groups/dynamics/abstracts/InternetEcology/options.html)
7. Bernardo A. Huberman and Rajan M. Lukose, **Social Dilemmas and Internet Congestion** [[abstract]](http://www.parc.xerox.com/spl/groups/dynamics/abstracts/InternetEcology/storms.htm)
8. Bernardo A. Huberman, Peter L.T. Pirolli, James E. Pitkow, and Rajan M. Lukose, **Strong Regularities in World Wide Web Surfing**, Nature ??? [[abstact]](http://www.parc.xerox.com/spl/groups/dynamics/abstracts/InternetEcology/surfing.html)
9. William W. Cohen, Wei Fan, **Web-Collaborative Filtering: Recommending Music By Spidering the Web**, WWW9 [[html]](http://www9.org/w9cdrom/266/266.html)
10. Mike Perkowitz Oren Etzioni, **Towards Adaptive Web Sites: Conceptual Framework and Case Study**, WWW8 [[html]](http://www8.org/w8-papers/2b-customizing/towards/towards.html)
11. Marc Langheinrich, Atsuyoshi Nakamura, Naoki Abe, Tomonari Kamba, Yoshiyuki Koseki, **Unintrusive Customization Techniques for Web Advertising**, WWW8 [[html]](http://www8.org/w8-papers/2b-customizing/unintrusive/unintrusive.html)
12. Jim Hendler, **Is there an intelligent agent in your future?** Nature, Web matters, 11 March 1999. [[html]](http://helix.nature.com/webmatters/agents/agents.html)
13. Steve Kirsch, **The future of Internet search (keynote address)**, SIGIR 1999 [[pdf]](http://www.acm.org/pubs/articles/proceedings/ir/290941/p1-kirsch/p1-kirsch.pdf)
14. G. Pandurangan, P. Raghavan, and E. Upfal. Using PageRank to Characterize Web Structure, Proceedings of the 8th International Computing and Combinatorics Conference (COCOON), 2002.
15. [G. Pandurangan](http://www.cs.brown.edu/people/gopal) , [P. Raghavan](http://theory.stanford.edu/people/raghavan/index.html) , and [Eli Upfal](http://www.cs.brown.edu/people/eli), [**Building Low-Diameter P2P Networks**](http://www.cs.brown.edu/research/webagent/focs-2001.pdf). Proceedings of the 42th IEEE Symp. on Foundations of Computer Science. 2001.
16. S.R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and [Eli Upfal](http://www.cs.brown.edu/people/eli), [**The Web as a graph**](http://www.cs.brown.edu/research/webagent/pods-2000.pdf). Proceedings of the 19th ACM Symposium on Principles of Database Systems, pp 1-10, 2000.
17. Thomas Hofmann, [**Learning Probabilistic Models of the Web**](http://www.cs.brown.edu/research/webagent/Hofmann-SIGIR00.pdf), ACM SIGIR 2000
18. R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and [Eli Upfal](http://www.cs.brown.edu/people/eli), [**Stochastic models for the Web graph**](http://www.cs.brown.edu/research/webagent/focs-2000.pdf). Proceedings of the 41th IEEE Symp. on Foundations of Computer Science. 2000.
19. <http://en.wikipedia.org/wiki/Trident_%28layout_engine%29> , downloaded 1/9/2009
20. <http://msdn.microsoft.com/en-us/magazine/cc163723.aspx> , downloaded 1/9/2009

**1.5 Overview of the document.** Provides a birds-eye view of what information is given in this report, and where in the report it can be found. Description can be focused towards different types of reader, e.g. end-user, technical, developer, specialist, domain expert, accountant, legal, management, customers customer etc.

**2 General Description**

**2.1 Relation to current projects.**

This project is heavily dependant on the open source PBX called asterisk [[1]](#footnote-1)as a reference PBX which means that the entire test will be conducted exclusively on it (for call switching but especially for simulation clients registration test where the asterisk PBX will be the registration target.)

Another aspect is the sip UA component that will be taken from the PJSIP project and integrated into the simulation client.

**2.2 Relation to predecessor and successor projects.**

NA – this is a new project.

**2.3 Function and purpose.** Describes the main functions the product must perform, gives an overview. (Details are set out in Section 3.) Takes a developer-centric approach.

|  |  |
| --- | --- |
| Function | purpose |
| Test script creation | Give a facility to create test scripts that will be distributed to test clients and performed by them |
| Bulk SIP registration | To implement the registration performance test of many clients to the PBX |
| SIP call setup | To implement the call performance test there is a need for the ability to generate calls |
| Central synchronization point | In order to gather and display test results, broadcast test scripts, generate reports and sync the test, there is a need for the central synchronization point implemented by the management server. |
| SIP call grading | In order to give a grade for the success of the test there is a need to aggregate and analyze the test result data into one grade, Implemented both in client (QoS measurer component) and server (Result Analyzer) |

Table

**2.4 Environmental considerations.**

The product will be used at the client site and its components (Management server and simulation client PCs) will be distributed across the customer network. Actually the management server will reside at the customer HQ where as the simulation clients will be carefully placed at the customer network nodes (i.e. client per switch/router at the customer premises), in parallel to employee PC (it can run as an application on the customer employee’s in the background).

The system is designed to be managed by an IT personnel who has knowledge in VoIP protocols who will plan perform and analyze the tests.

The product is designed to run on windows XP OS.

**2.5 Relation to other systems.**

A complete VoIP system which is based on the SIP protocol consists of several entities connected by an IP network. The entity which performs the ‘phone’ functionality is called SIP User Agent (SUA) which internally is composed of User agent Server (UAS) which is responsible on accepting calls and a User agent Client (UAC) which is responsible on initiating calls. The SIP PBX contains 2 SIP entities: the Proxy and the Registrar. The Registrar role is to keep record on every SUA in the system and the Proxy role is to mediate between clients during the call setup.

As seen from the diagram simulation client (which act as SUA’s) will be distributed in strategic locations across the customer network and will be administered using the management server. Also the system will test the customer PBX in terms of client registration performance and call switching ability (where the PBX will act as SIP proxy).

Context diagram 1

IP Network

SIP UA



SIP UA



Simulation Client

Simulation Client

SIP PBX

Proxy

Registrar

Management Server

**2.6 General constraints.**

* As the system is based on consumer PC hardware the system will be limited in concurrent calls that can be setup and analyzed due to the limited CPU resources of a PC. The precise number of concurrent calls will be determined via trial and error during the coding phase. It is estimated that about 255 concurrent calls can be achieved.
* The SIP UA is constrained by the stability of the open source SIP library it is based upon.

**2.7 Model description.**

The following diagram will describe the data flow between the various processes in the system.

In this top level diagram one can see the test parameters (chosen in the GUI of the management server) is sent to the test creation process which generates a set of test steps objects. These objects are converted into a test script which is sent to the simulation client. The simulation client reads the script and generates a test sequence that controls the test run. The raw results of the test run are sent to be analyzed, and the analyzed results (i.e. packet loss, jitter, delay, voice quality grade) are ordered into a report which will be visible to the user.

DFD Top Level

Figure 1

Test steps

Analyzed results

Send data

Results

Test sequence

Client data

Client list

D1

Test Data

D2

Test script

D2

Simulation clients

D3

Report

DFD-1 Test Creation:

The simulation clients register to the management server which records them in the simulation clients DB. The user is presented with a test creation form (at the management server) where all the registered simulation clients are displayed. The user then selects the test type (registration, call setup performance, voice call quality), and the registered clients which will take part in this test and matches them in pairs.

The test parameters are translated to ordered and timed test steps in a test script.

Figure

Test parameters

Test Sync

Retrieve client data

Update client

D3

Simulation clients DB

Client parameters

D1

Test form

D2

Simulation client

D4

Test script

DFD-1 Test Run:

The test script is sent to the clients which run the test according to the script. Results are collected by the client during the test run, analyzed and sent back to the management server. The server builds a report out of these results, and stores it in the result DB.

Figure 3

Result list

Result storage

Send script to clients

D3

Result DB

Result Collection

D2

Test results

D1

Test script

DFD-1 Result analysis:

The management server extracts the result set and format them into analyzed reports (for example turn time stamps into time duration) and displays them to the user in a report form.

Figure 4

Analyze results

Result retrieval

Format results

D1

Result DB

Display results

D2

Analyzed results

D3

Display form

**3 Specific Requirements**

**3.1 Functional requirements.**

Management server:

Figure 5

GUI

Result Collector

Result Analyzer

Message Processor

Test Scheduler

Simulation Client Manager

* GUI: this module will be the interface to test creation – that is utilising it one can:
  + Select test type
  + Control test state (start, end, cancel, reset)
  + Display of test results.
* Result Analyzer: this module analyzes the collection of test results gathered from the simulation clients, it’s output depends on the test type:
  + Success rate of registrations/call setup
  + Quality grade of calls
* Simulation Client Manager: this module manages the simulation clients registration. It handles client state records and keeps track on them.
* Test Scheduler: this module manages the test synchronization. For example putting one simulation client in wait for call state while setting another client to initiate call to the previous one.
* Result Collector: this module is responsible of gathering test results from the message stream and sending them to be analyzed.
* Message Processor: this module sends and receives messages to/from simulation clients at one end and transferring them to the appropriate module at the other side.

Simulation client structure:

Logger

Client Control

QoS Measurer

Message Processor

SIP UA

Script Manager

Figure 6

* Simulation Client Control: this module handles message transfers with the management server.
* SIP UA: this module is where the call is actually made. It simulates a phone through which calls to other customers can be setup, answered, and maintained.
* QoS Measurer: this module processes the voice stream and gives it a quality of service grade (the amount of match between a saved copy of the original voice stream and the actual voice stream received).
* Script Manager: this module is responsible of running the test script by manipulation of the SIP UA module.
* Logger: this module is responsible of registering all actions and states of the simulation client for debugging purposes.
* Message Processor: this module is responsible of sending/receiving message through the network to/from the management server.

**3.2 Performance requirements.**

The system shall handle up to 255 concurrent calls [estimation].

The system shall manage up to 32 simulation clients [estimation].

**3.3 Interface requirements.**

Communication across the network between simulation client and management server will be using messages formatted as XML strings (SOAP).

Following are UML diagram of the management server and simulation clients:[[2]](#footnote-2)

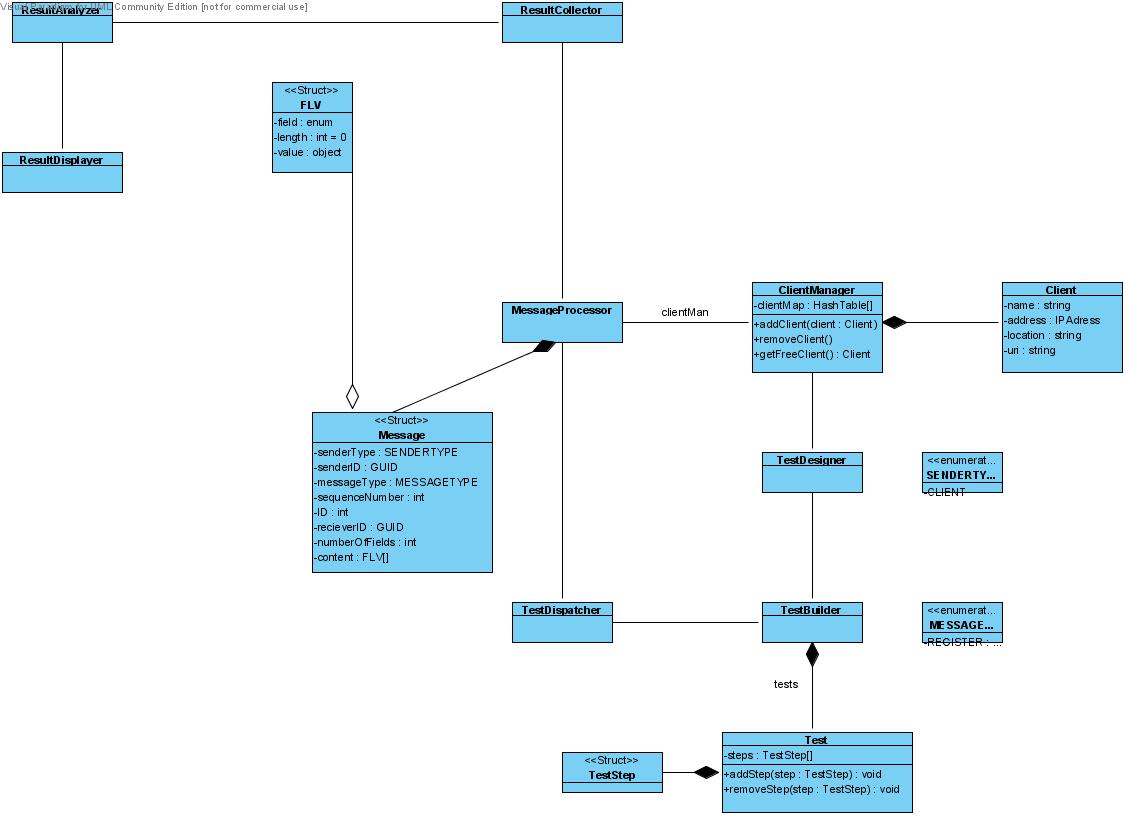


Figure -Management server

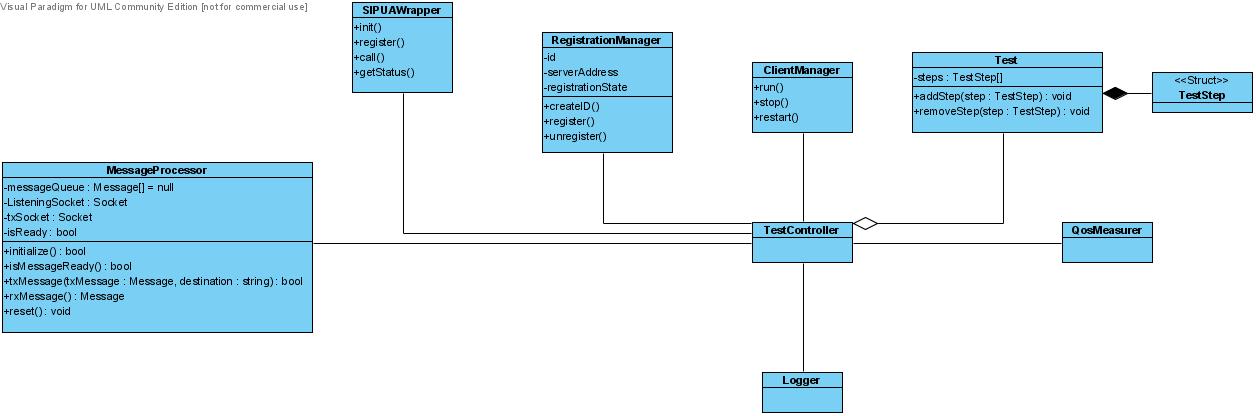


Figure - Simulation Client

**3.4 Resource requirements.**

Platform: Pentium III and above with 512MB for simulation clients and management station.

OS: Microsoft Windows XP sp2.

Network: this is actually one of the systems under test.

PBX: linux machine running Asterisk PBX software.

Development platform: Visual studio 2005.

**3.5 Verification requirements.** As in test plan document

**3.6 Security requirements.** NA

**נספח ב2 - (Software Test Plan (STP**

**(מבוסס על IEEE Std 829-1998)**

1. **Introduction**

This test plan will verify the entire requirements in the PBX tester SRD were implemented correctly. The philosophy of testing is to test each component as isolated as possible (Unit test each component of simulation client and management server by building a test class for each component class) and then perform tests on the system as a whole.

Because this system is unique in terms of that it is itself a testing platform there is a need to verify the correctness of the results it generates and especially the indexes and measures it makes. To that end each index (jitter, delay, packet loss etc’) to be tested will be compared against an expected synthetic data known to generate a certain value.

The availability of such data is incomplete at this time.

1. **Test items** 
   * + simulation client
     + management server
2. **Features to be tested**

* Test functionality of simulation client:
  + Register client to management server
  + Register client to Registrar
  + Make call from client
  + Make call to client
* Test functionality of management server
  + Create test script using GUI
  + Register client to management server
  + Dispatch script to client
  + Parse dummy result set
  + Display dummy Result
* Test performance of registration
  + Make incremental rate registrations to Registrar
* Test performance of call setup
  + Make incremental rate call setup between 2 clients
* Registration test creation.
  + Create Registration test script using GUI
* Call setup test creation without proxy.
  + Create Call setup w/o proxy test script using GUI
* Call setup test creation with proxy.
  + Create Call setup with proxy test script using GUI
* Simulation client registration.
  + Inject Client Registration message to server.
* Test dispatch.
  + Dispatch all the above test script and capture them with a sniffer.
* Registration test execution by simulation client.
  + Inject Registration script to client and execute it.
* Call setup test execution by simulation client.
  + Inject call setup script to client and execute it.
* Result Collection and analysis.
  + Run analysis on a perfect voice file and check the result indexes are perfect
  + Create a voice file with known jitter and delay using audio editing tool and verify indexes reflect these anomalies.
  + Take a recorded voice packet stream and cut several packets out to simulate packet loss. Fid it to the analysis module and verify it reports correct packet loss.
* Result Display.
  + Feed results to display module and verify they are displayed correctly.

1. **Features not to be tested:**
   1. the open source SIP library.

1. **Environmental needs –** as in project proposal section 4.
2. **Schedule** – as in Gantt chart.

1. Asterisk is an open source/free software implementation of a telephone private branch exchange (PBX) originally created by Mark Spencer of Digium. Like any PBX, it allows a number of attached telephones to make calls to one another, and to connect to other telephone services including the public switched telephone network (PSTN). [↑](#footnote-ref-1)
2. This is an initial and general UML diagram and probably will change in down the development road [↑](#footnote-ref-2)