https://www.google.com/a/cpanel/guttman.ort.org.il/images/logo.gif?service=google_default

**מערכת גישה אוטומטית לאתרים**

**מבוססת SSO**

*מסמך עיצוב פרויקט*

***אליזבת לנגרמן***

ת.ז: 208568675

בית ספר: אורט גוטמן נתניה

מורים מנחים: מיכאל צ'רנובילסקי

גד רוזנטל

תאריך הגשה: 02.2016

**מבוא**

במסמך זה ארחיב על מבנה הפרויקט שלי. אסביר כיצד הפרויקט יהיה מעוצב- מאילו חלקים יהיה בנוי וכיצד ייראה ממשק המשתמש.

המוצר שאני מפתחת אמור לתת למשתמש מסויים אפשרות גישה לאתרים שונים ללא צורך בסיסמאות רבות. המשתמש יוכל להתחבר למערכת (באמצעות סיסמא אחת בלבד) והיא תאפשר לו להתחבר לשירותים שתומכים במוצר זה ללא צורך בשם משתמש וסיסמא ייחודיים. כלומר, במקום להתחבר כרגיל, המשתמש יבחר באופציה המתאימה ויקבל גישה לחשבון שלו באופן אוטומטי.

**המסמכים הרלוונטיים**

[1] - מסמך הצעת ותיאור הפרויקט.

**ארכיטקטורת המערכת**

הפריקט שלי מורכב משני שרתים מרכזיים: השרת הראשי ()IDP, ושרת לדוגמא שיתמוך באפשרות שאני מפתחת (SP). בשני השרתים תהיה תמיכה לחיבור לקוחות במקביל- multithreading.

*השרת הראשי(IDP) - חלוקה למודולים:*

* **New User Registration**

מודול שאחראי על הרשמה של משתמשים חדשים, בין אם לשרת SP מסויים או לשרת המרכזי- IDP בעצמו.

* מקבל מזהה של הSP שממנו נשלחה הבקשה
* מקבל ID של המשתמש כפי שנרשם בשרת ה SP ושומר אותו
* מבקש פרטי הרשמה או התחברות מהמשתמש
* בודק האם המשתמש קיים, או שהוא חדש לחלוטין
* בהתאם, שומר את המידע החדש שנוסף על משתמש זה
* Redirect בחזרה לשרת הראשון (SP) + הודעת הצלחה
* **User Login**

מודול שתפקידו לאפשר למשתמש להתחבר לשרת הראשי ומספק את פרטי ההתחברות של המשתמש לSP.

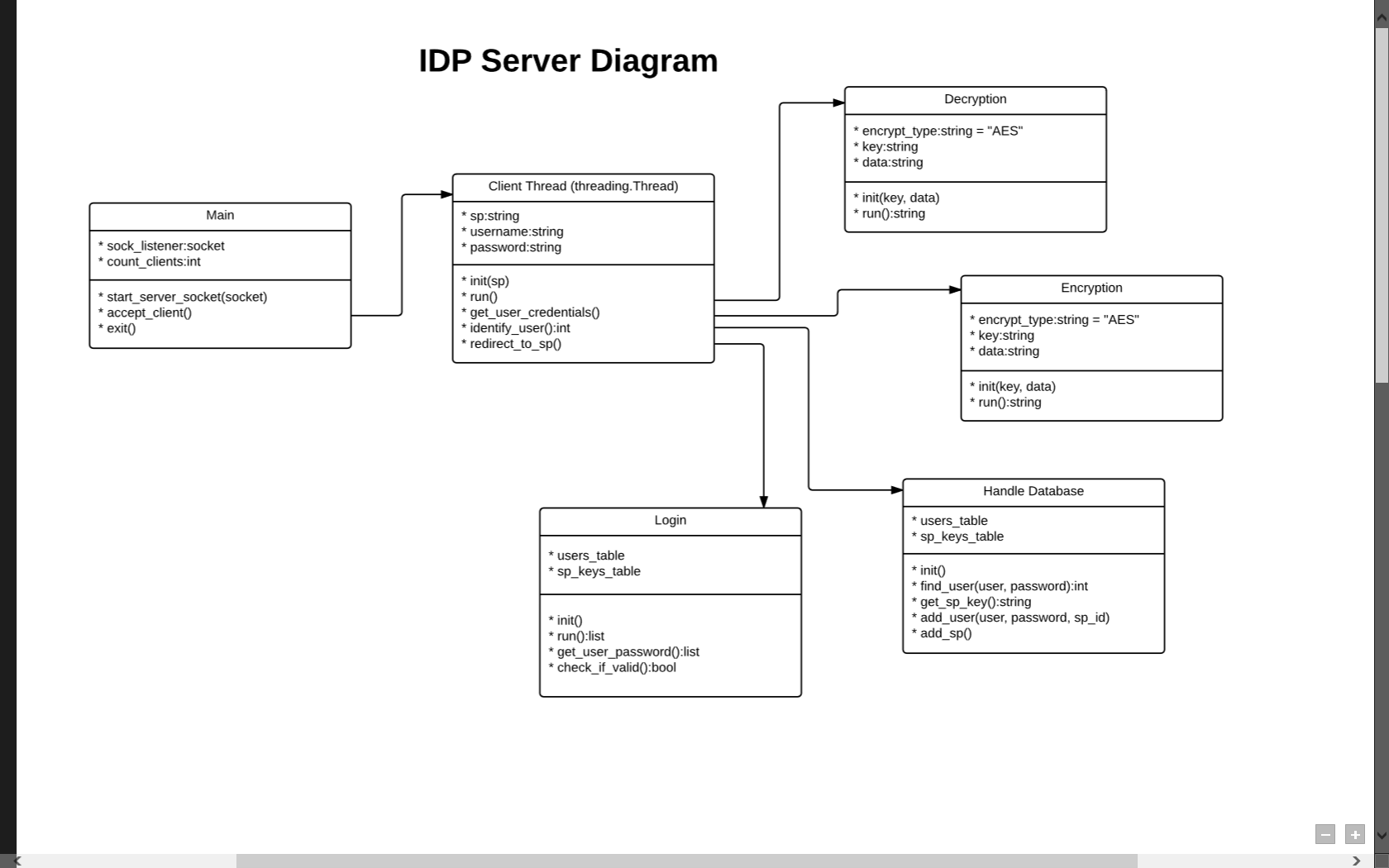
* מקבל מזהה של הSP שממנו נשלחה הבקשה לזיהוי לקוח
* מבקש מהמשתמש להכניס פרטי זיהוי
* מוודא את קיום המשתמש (בהתאם לDatabase)
* ממפה בין המשתמש לבין ה ID שלו בשרת ה SP המבקש
* Redirect בחזרה לשרת הראשון (SP) + שליחת user SP ID ל SP.
* **SP Registration**

מודול שתפקידו תקשורת ראשונית עם שרת שמתחיל להשתמש במערכת.

* מקבל בקשה משרת הSP החדש
* מבקש ממנו פרטים חיוניים לתקשורת ביניהם (לדוגמא מידע כללי, הכתובת שאליה לעשות redirect, רשימת מזהים של המשתמשים הרשומים לשרת זה)
* מנפיק מפתח להצפנה סימטרית בין שני הצדדים ושולח לSP.
* **Handle Database**

המודול מכיל את מסד הנתונים ואת הכלים לעבודה מול מסד הנתונים.

* אפשרות הוספת משתמש
* חיפוש משתמש והחזרת ה ID שלו בהתאם לSP המבקש
* חיפוש SP ואפשרות להוספה



*השרת המשתמש במערכת(SP) - חלוקה למודולים:*

* **Login**

מודול המאפשר למשתמש כלשהו להתחבר לשרת.

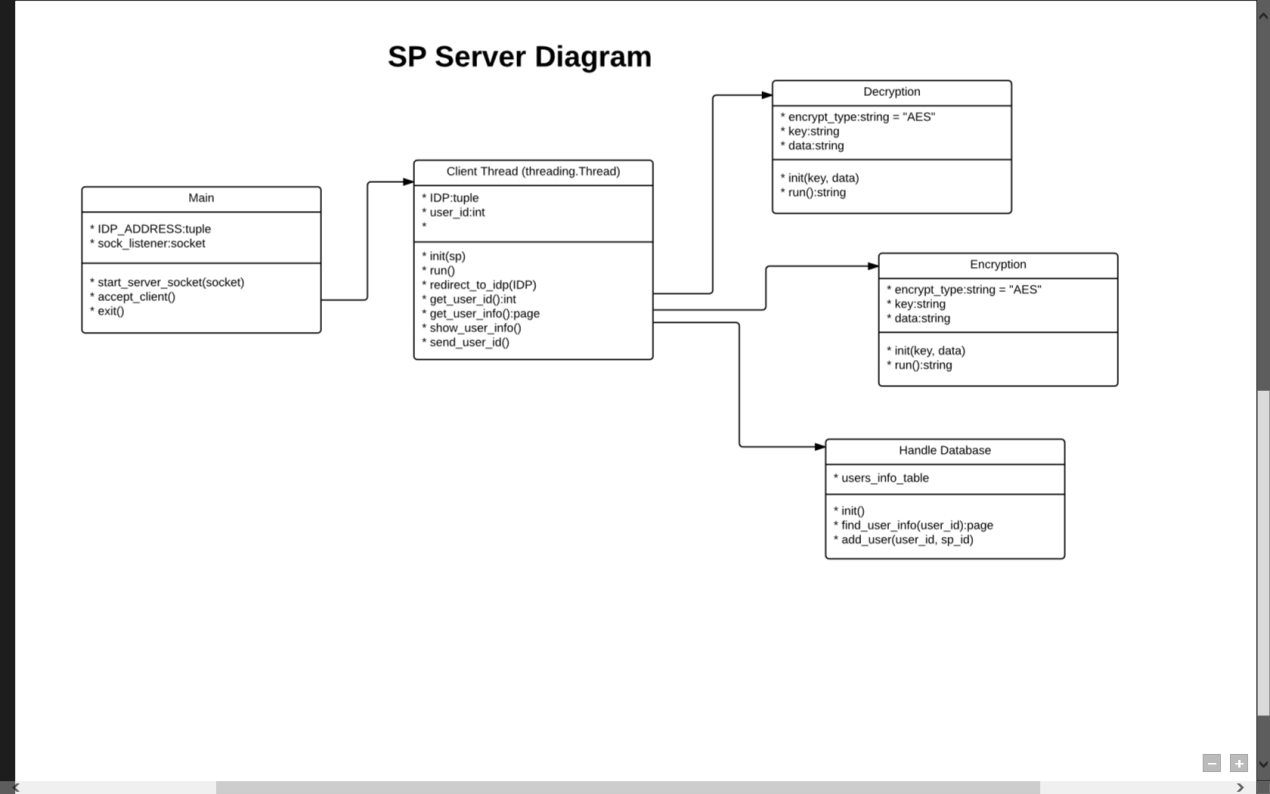
* אפשרות התחברות באמצעות שם משתמש וסיסמא או באמצעות שרת ה IDP.
* Redirect לשרת המרכזי IDP + שליחת מזהה ייחודי מוצפן של הSP.
* הצגת נתוני המשתמש אחרי ההתחברות, כרגיל.
* **Register**

מודול שאחראי על הרשמת משתמש חדש לאתר, והתקשורת עם השרת הראשי במצב זה.

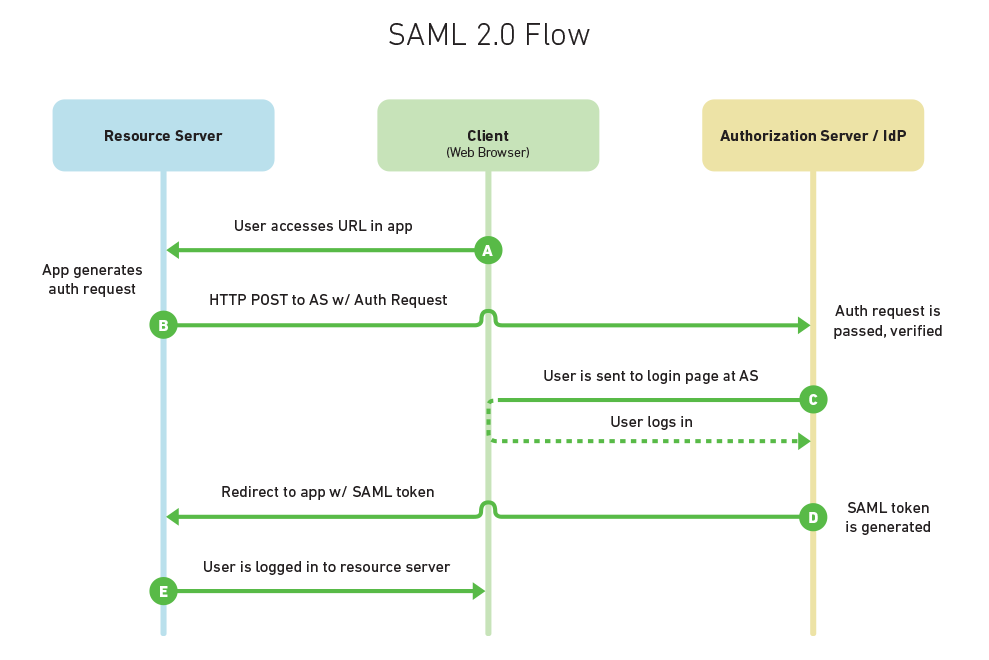
* Decrypt
* Decode
* במקרה של הצלחה, Redirect למודול User Page
* **Handle Database**

המודול מכיל את מסד הנתונים ואת הכלים לעבודה מול מסד הנתונים.

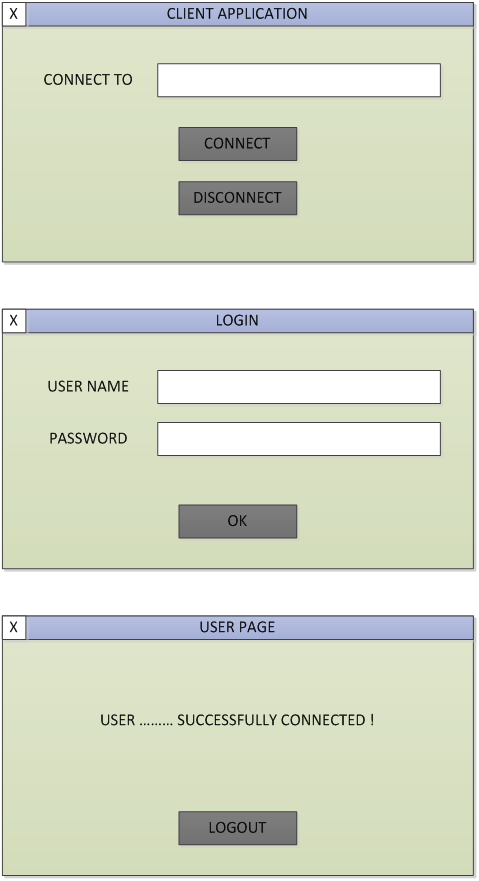
* אפשרות הוספת משתמש
* חיפוש משתמש והחזרת העמוד הפרטי שלו בהתאם ל userID



המערכת מבוססת על מגנוןSAML , שמבנהו כבתרשים:

 <https://www.mutuallyhuman.com/blog/2013/05/09/choosing-an-sso-strategy-saml-vs-oauth2/>

**ממשק משתמש**



**תהליך מחקר ופיתוח**

המערכת שאני בונה בפרויקט זה מטרתה יצירת אפשרות SSO. יש מספר שיטות ליצור זאת, ולאחר מחקר על SAML ועל 2OAUTH, החלטתי שהמבנה שלSAML מתאים יותר למבנה המערכת שאני רוצה לפתח.

התחלתי לחפש מודול שיסייע לי ביצירת מערכת שפועלת לפי דוגמת SAML. מצאתי מספר מודולים, ומתוכם בחרתי את pysaml2. רציתי להשתמש תחילה במודול זה משום שהוא מאפשר לעבוד עם SAML, ליצור IDP, SP וכו'.

התחלתי ללמוד את השימוש במודול, אך לאחר מחשבה מעמיקה הבנתי שאעדיף לא להשתמש בו, אלא אוכל לבנות מערכת שצורת פעולתה דומה למבנה המוגדר בSAML, ללא שימוש במודול מיוחד לשם כך.

תיארתי מבנה מערכת חדש, שבו לא יהיה צורך במודול זה. שרטטתי גם תרשים זרימה שמתאר באופן רציף את תהליך פעולת המערכת. פירטתי בתרשים מה תבצע כל אחת מהתוכנות שהן חלק מהפרויקט שלי.

באמצעות התרשים התחלתי לכתוב את הקוד בצורה מסודרת. ראשית התחלתי מכתיבת הService Provider כולל ממשק המשתמש שלו. לאחר שזה מוכן והגרסא הראשונית שלו פועלת כראוי, התקדמתי לכתיבת הגרעין של המערכת – ה Identity Provider.

לאחר שבניתי מערכת הכוללת ממשק #C, הבנתי שעדיף יהיה לבנות את הפרויקט כך שהשרתים יהיו שרתי web. כלומר, הלקוח יהיה הדפדפן- באמצעות הדפדפן יפנה המשתמש אל שרת(SP) כלשהו, שיחזיר לו redirect לשרת ה IDP. שיטה זו נפוצה ויעילה יותר לדעתי מאשר אפליקציות בודדות עם ממשק #C, גם מפני שאפשר להפעיל שרת ולהתחבר אליו ממחשבים שונים באמצעות הדפדפן.

באמצעות תרשימים לתהליכים שארצה לממש בפרויקט שלי, הגדרתי סופית את המבנה והפרוטוקולים כאשר מדובר בשרתי web. במהלך בניית התרשימים נתקלתי במספר שאלות:

* *באחד מתהליכים אלו, אמור שרת הIDP להפנות את הלקוח בחזרה לשרת הSP. כיצד יפנה אותו בחזרה לשרת הנכון?*

ה SP צריך לשלוח לIDP מידע בנוגע לזהותו. דרך אחת לעשות זאת היא באמצעות שדה הhttp referrer. דרך נוספת היא באמצעות שליחת מזהה ייחודי של השרת המבקש, עם שליחת redirect. הדרך השלישית היא שליחת המזהה בנפרד, ישירות לשרת הIDP. אמנם בדרך זו המידע לא עובר דרך המשתמש ולכן התקשורת מאובטחת יותר, אך יש צורך בתקשורת נוספת עם השרת המרכזי, שעלולה להיות לא מתואמת מבחינת זמנים ונוחות. בחרתי להשתמש בדרך של שליחת מזהה מוצפן עם שליחת הredirect.

* *באיזה סוג הצפנה כדאי להשתמש?*

ראשית חשבתי להשתמש בהצפנה א-סימטרית, משום רמת האבטחה הגבוהה יותר. בהצפנה סימטרית יש צורך בהעברת מפתח, ולכן יורדת מידת האבטחה. אבל, במבט מעמיק יותר, שני הצדדים צריכים להצפין ולפענח מידע במערכת זו. לכן, יהיה יעיל יותר להשתמש בהצפנה סימטרית בכל זאת.

* *מתי ליצור וכיצד להעביר את מפתח ההצפנה בין שני השרתים?*

עבור כל שרת SP שמתחבר אליו, נחוך ל IDP מפתח קבוע. לכן, צריך להיות מנגנון של רישום SP, שבמהלכו יחליפו פרטים שני השרתים, וגם את מפתח ההצפנה. כך אין צורך בהחלפת מפתחות הצפנה בזמן שלקוח מתחבר, וגם לא בכל פעם מחדש.

* *כיצד לשמור את המידע על המשתמשים הרשומים?*

לצורך שמירת שמות המשתמשים והסיסמאות אשתמש ב SQLite3. כך גם לצורך התאמת מזהה המשתמש בין השרתים ולשמירת מידע של כל משתמש.

* *איך אבנה שרת web?*

התחלתי ללמוד את השימוש במודול flask לפייתון. מודול זה מאפשר בניית שרת web בשפת פייתון.

**ניהול פרויקט עתידי**

|  |  |
| --- | --- |
| כתיבת שני השרתים עפ"י המבנה | **10.2** – 27.2 |
| השלמת כתיבת database  +  השימוש בו בכל שרת | 28.2 – **9.3** |
| בדיקה ותיקון שגיאות,  ותוספות במידת הצורך | 9.3 – 26.3 |
| תוספות אחרונות לפרויקט  +  תיק העבודה הנלווה | 27.3 – **6.4** |
| תוספות אחרונות, חזרה והכנה לקראת הגשת הפרויקט | 6.4 – 1.5 |