



Web solution for viewing knowproj videos

ספר פרויקט

רכזת פרויקטים : ד"ר ענת אהרוני

מנחה הפרויקט : מר יניב לביא, אינטל

מגישים :

דוד רייספלד, ת.ז. 310338694

גלית בלמס, ת.ז. 200663029

תאריך : 13.9.2015, סמסטר ב' תשע"ה

תוכן עניינים:

1.	מבוא	2
2.	מסמכי הגדרת הפרויקט	3
2.1.	הגדרת הפרויקט	3
2.2.	דיאגרמת הקשר בין שלושת הפרויקטים	6
2.3.	לוח זמנים	7
3.	אפיון הפרויקט	9
3.1.	סביבת פיתוח	9
3.2.	תוכנית העבודה	9
3.3.	מסמך דרישות לקוח	10
3.4.	הגדרת המשימות	15
4.	תיאור התהליכים המרכזיים והאלגוריתמים	16
4.1.	אלגוריתם מציאת flow אופטימלי מתוך עץ התלויות	16
4.2.	תהליכים מרכזיים	17
5.	צילומי מסך עיקריים של המערכת	18
6.	מסמכי בדיקות	21
7.	סיכונים טכניים וקשיים למימוש דרישות הלקוח והדרך להתגבר עליהם	31
8.	עמידה ושינויים בדרישות לקוח	33
8.1.	עדכוני לו"ז של תהליך הפרויקט ועמידה בדרישות הלקוח	33
8.2.	שינויים בדרישות הלקוח	34
9.	לקוח סיכום דוחות חודשיים ומפגשים	35
9.1.	סיכום הדוחות החודשיים והפגישות עם הלקוח	35
10.	סיכום הפרויקט ומסקנות	36
	ביבליוגרפיה	37
	נספחים	38

1. מבוא

פעמים רבות עובדים המגיעים למקום עבודה חדש מוצאים את עצמם מבולבלים ואבודים בתוך ערימת המידע החדש המגיע אליהם מכל עבר – מונחים שונים הנשמעים מכיוונם של עמיתיהם לעבודה, מנהליהם, לקוחות, שמות של כלים, מכשירים, תוכנות ועוד.

מערכת Knowproj נועדה לנהל תכני לימוד וסרטוני וידאו, המוצגים בצורה מסודרת וידידותית למשתמש – הן לעובדים חדשים המנסים לעשות סדר במידע החדש והן לעובדים ותיקים, שכן כמות המידע היא עצומה ולעולם לא מפסיקים ללמוד.

מערכת Knowproj גם מאפשרת למנהלים ליצור Flow (המושג יתואר בפירוט רב בהמשך הספר), איזשהו רצף של מידע, בו הוא מעוניין שעובדיו יצפו וירכשו את הידע הטמון בו.

בתור עובדת אינטל בעצמי (גלית), נזכרתי בימי הראשונים, וחשבתי לעצמי שמערכת כזו הייתה יכולה להיות שותפה נהדרת בצעדי ההתחלה וההיכרות עם המערכת והמידע הרב שהיה עליי ללמוד. לכן, לאחר שיחה קצרה עם מנחה הפרויקט והבנת רעיון המערכת – החלטנו להירתם למשימה ולקחת את הפרויקט כפרויקט הגמר שלנו.

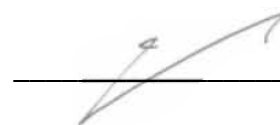
2. מסמכי הגדרת הפרויקט

2.1. הגדרת הפרויקט

שם סטודנט א :	גלית בלמס	תז :	200663029
שם סטודנט ב :	דוד רייספלד	תז :	310338694
שם הפרויקט :	Web solution for viewing knowproj videos		
מנחה : יניב לביא			
טלפון :	נייד : 7884056-054		
	Email: yaniv.lavi@intel.com		

חתימת

המנחה :



KnowProj is an experimental project to manage training/teaching content. The current requirements document refers to a web application that enables a user to view training material supplied by the system, grade it, keep records of how he benefits from the training, and use this records to get more accurate training material on his next interaction with the system.

Our project is a part of 3 projects that assembling together the final project:

1. Server side
2. Client side
3. Lerner client-server side (our project)

A diagram showing the connections between 3 projects is attached In the end of this document.

מטרת הפרויקט :

AJAX, PHP , SQL server
 Wampserver- learn how to use
 Development environment for JAVA and
 PHP - Eclipse

דרישות קדם :

- .1 Create work platform – server, client, browser display. (using Wampserver)
- .2 To enable each team to get a progress, "dummy requests" and "dummy results" need to be provided as first step
- .3 Get the Input from learner (our user) to start work flow
- .4 Create a "message", containing the learner requests, to send the main server (not our learner-server) containing the learner requests
- .5 Get the answer returned from the main server (the learn flows)
- (server2server : Get tree recursive just for kb->delivery)
- .6 Translate the learn flows so we can use them
- .7 Choose up to 3 flows, Sorted (by algorithm and user selection)
- .8 Show learner the flows, gray out things the user know. User will choose a flow.
- .9 Enable watching deliveries one by one
- .10 Enable marking kbits
- .11 for each delivery - show it's data
- (server2server - Get tree single depth (first level) for any links)
- .12 for each element – show it's data
- .13 add flows options:
 - past flows
 - wishlist flows
 - personally defined flows

שלבי ביצוע
 תכנית עבודה:

<p>Work with data supplied by main server .14 (other teams)</p> <p>Work with real content .15</p> <p>Define set of tests to run on each version .16</p> <p>Define system demo .17</p> <p>(*In addition, a full and more detailed customer requirements document is attached.)</p>	
<p>First stage will be creating the interface .1 between server and client, have “dummy requests” and “dummy responses” so that server and client can work independently from that moment onwards.</p> <p>Perform Evaluation and planning of the .2 projects’ features, so we can understand, in 2 months, what will be in the final project.</p> <p>Build an initial skeleton (sheled??) to .3 illustrate technological capabilities (as cookie-saving, etc.).</p> <p>Architecture and design. .4</p> <p>Work with “dummy” content – supplied by .5 other teams (main server & client).</p> <p>Final stage – work with real content! .6</p>	<p>שלבים עיקריים ומטרות ראשיות :</p>
<p><u>Session</u>: must keep user's session and .1 history, so that user can get back to where he was if the window closes. Also, there should be an option to get to the same point in a work-flow through URL</p> <p>Learner should be able to search based on .2 things he saw/didn't see (“search ‘pattern’ only in videos I watched before”), and search in his own comments</p>	<p>שלבי בונוסים :</p>

We must use commercial apache server, have server-side in PHP, client side in JS, use AJAX for client requests from server through a set of defined APIs. DB is MySQL.

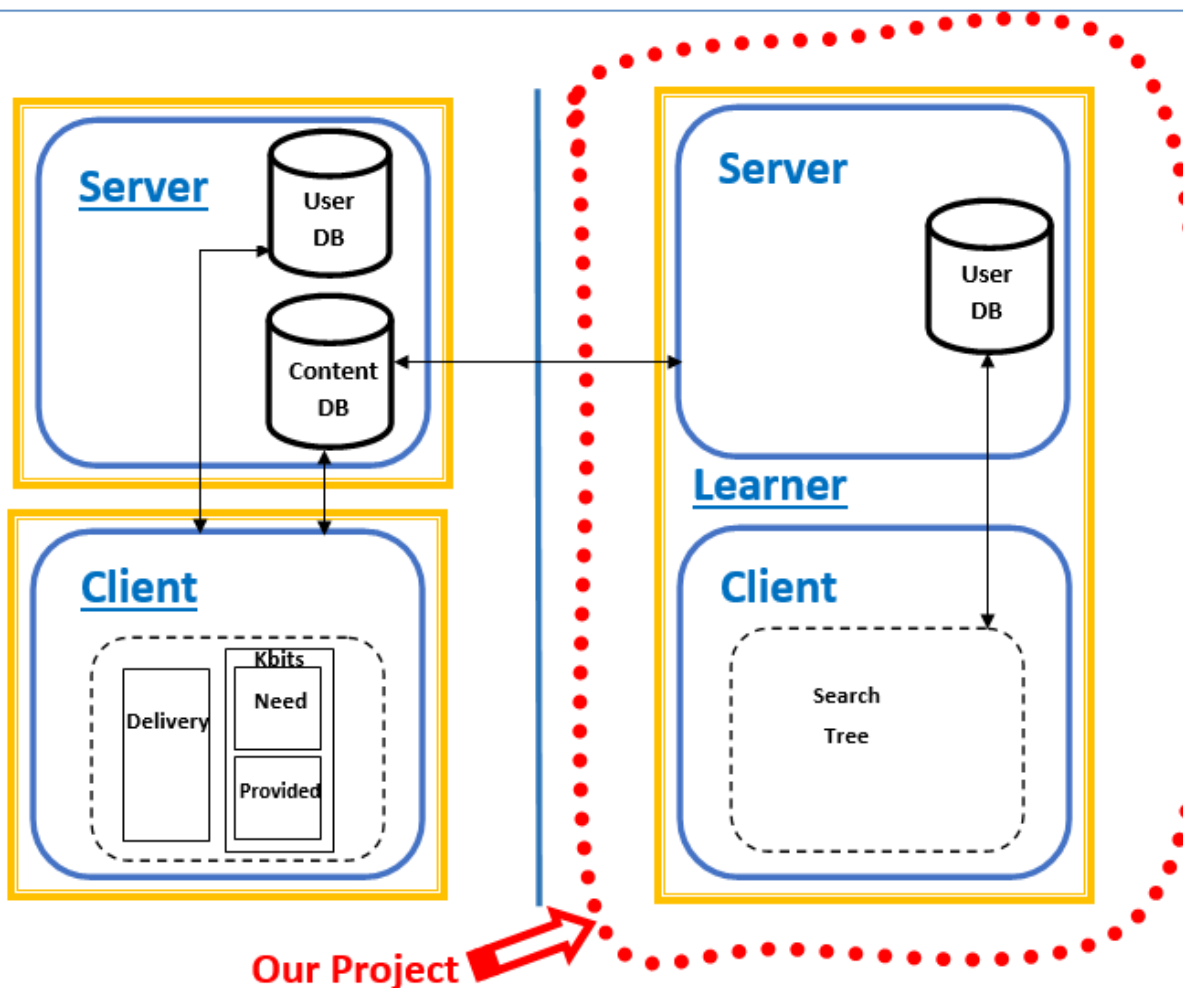
סביבה נדרשת :

Beginning of September 2015

קו סיום משוער :

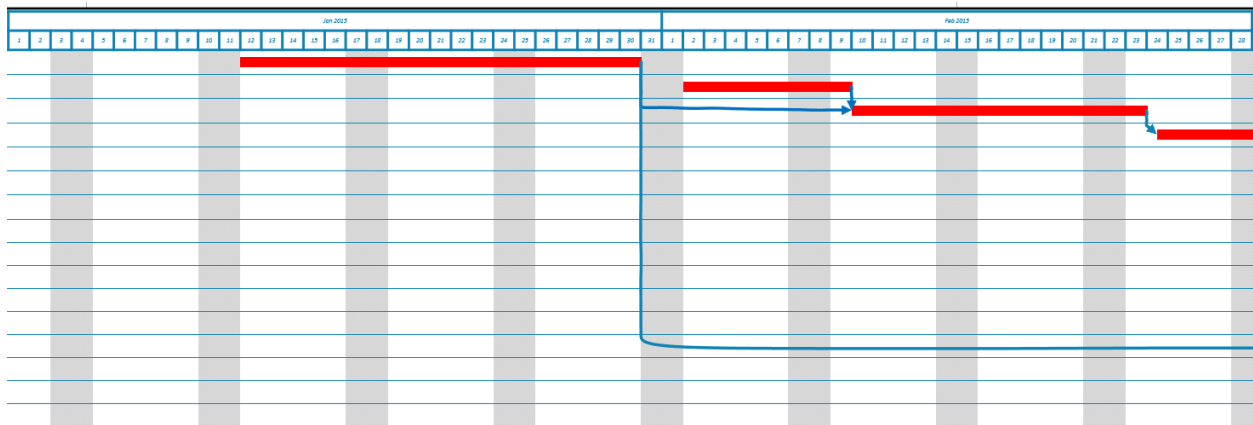
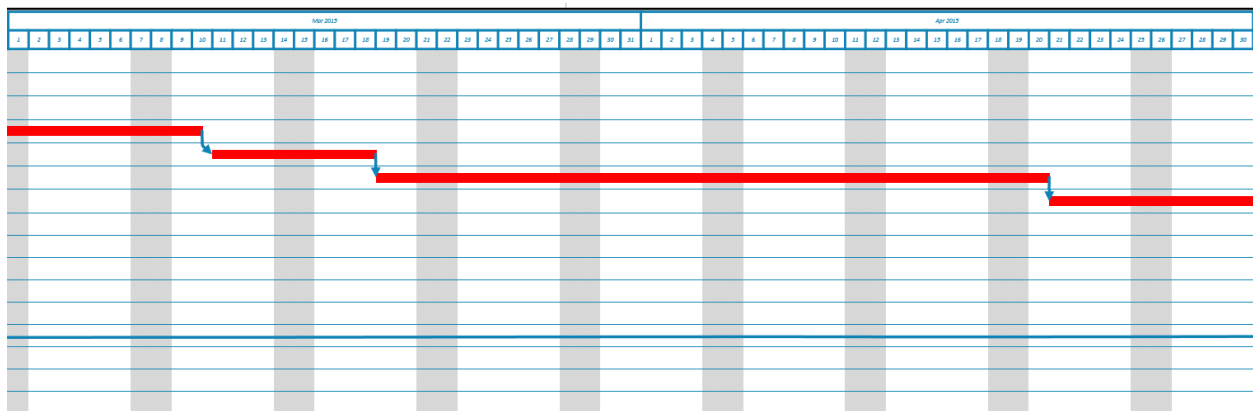
ספרות ומאמרים :

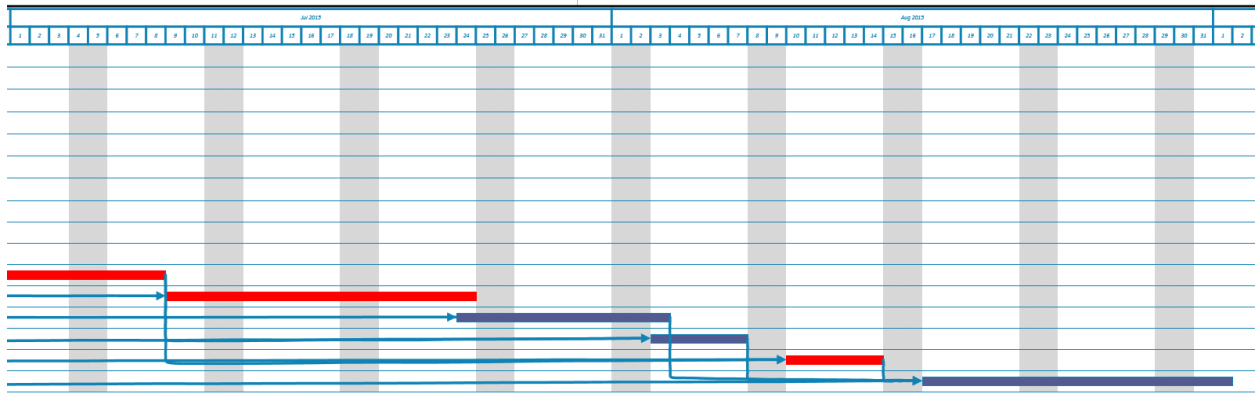
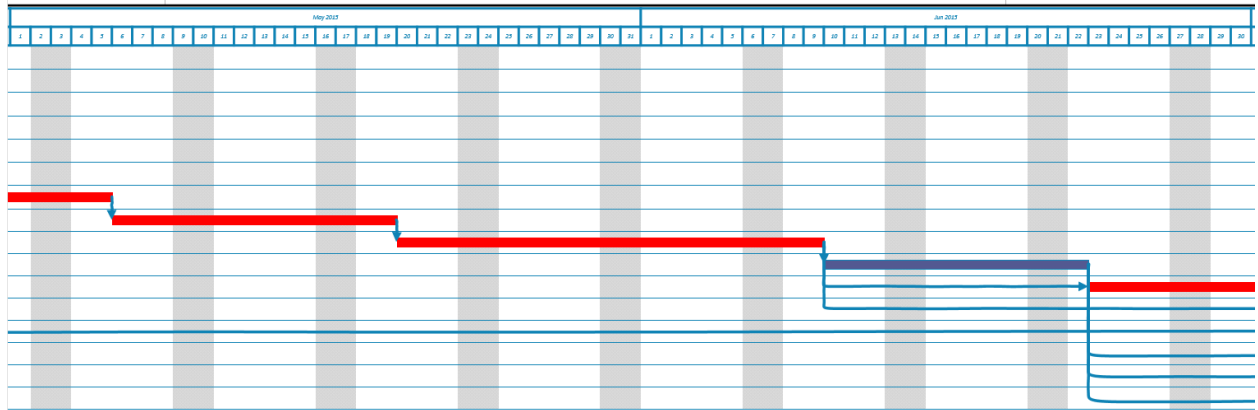
2.2. דיאגרמת הקשר בין שלושת הפרויקטים



2.3. תכנון לוח הזמנים

ID	Task Name	Start	Finish	Predecessors	Duration	Dec 2014										
						21	22	23	24	25	26	27	28	29	30	31
1	Create work platform	1/12/2015	1/30/2015		15d											
2	Dummy requests and results between teams	2/2/2015	2/9/2015	1	6d											
3	Get input from user-learner	2/10/2015	2/23/2015	1	10d											
4	Create the request and send it to main server	2/24/2015	3/10/2015	2,3	10d											
5	Get answer from main server	3/11/2015	3/18/2015	4	6d											
6	Translate answer into learn flows	3/19/2015	4/20/2015	5	23d											
7	Sort and choose up to 3 learn flows	4/21/2015	5/5/2015	6	11d											
8	Show learner the suggested learn flows	5/6/2015	5/19/2015	7	10d											
9	Enable watching deliveries one by one	5/20/2015	6/9/2015	8	15d											
10	Enable marking kbts	6/10/2015	6/22/2015	9	9d											
11	Show deliveries data (for each delivery)	6/23/2015	7/8/2015	9	12d											
12	Show element's data (for each element)	7/9/2015	7/24/2015	9	12d											
13	Add flows options: past, wishlist and personally defined	7/24/2015	8/3/2015	1	7d											
14	Work with data supplied by main server (other teams)	8/3/2015	8/7/2015	11,12	5d											
15	Work with real content	8/10/2015	8/14/2015	11,12	5d											
16	Define system demo	8/17/2015	9/1/2015	10,13,14,15	12d											





3. אפיון הפרויקט

3.1. סביבת פיתוח

- המערכת פותחה באמצעות השפות:
 - Php - פיתוח צד שרת (learner-server).
 - Java script - פיתוח צד לקוח (Learner-client).
 - angularJS - תקשורת צד שרת-לקוח. (Learner client-server).
 - Html5 - תצוגה - פיתוח התצוגה בדפדפן.
 - Css - עיצוב האתר.
- Eclipse – IDE.
- השרת בו השתמשנו הוא - Apache server
- מסד הנתונים – MySQL.
- השתמשנו בכלי חינוכי המכיל את השרת ומסד הנתונים הנ"ל – Wamp server.

3.2. תוכנית העבודה

1. לימוד שפות התכנות: php, java script, html5, css
2. הכרת סביבת העבודה – Eclipse
3. יצירת פלטפורמת העבודה – שרת, לקוח, תצוגה בדפדפן – באמצעות Wamp server
4. החלטה על פורמט הבקשות / תשובות שיועברו בין צד השרת (הפרויקט השני) וצד הלומד (הצד שלנו)
5. יצירת dummy requests ו dummy results בהתאם למבנה עליו הוחלט בסעיף 4
6. חיפוש – קבלת בקשת החיפוש מהמשתמש, הרכבת "הודעה" (עפ"י הפורמט מסעיף 4), שליחתה לצד השרת הראשי וקבלת התשובה בפורמט המוסכם (סעיף 4).
7. Flows – התשובה המתקבלת מהשרת הראשי הינה עץ תלויות "Dependencies tree" המכיל בתוכו את כל התלויות בין deliveries וה-kbits הנדרשים למען יצירת flow עבור בקשת המשתמש. בניית אלגוריתם החיפוש למציאת ה-flow האופטימלי נעשה בשלב זה.
8. בסוף התהליך יוצג שם ה-flow האופטימלי למשתמש, ולאחר לחיצה עליו יוצג רצף של deliveries למשתמש.
9. צפייה ב-deliveries שנמצאים ב-flows – במעבר עכבר על כל delivery ניתן לראות מה הסטטוס של המשתמש עבור delivery זה (נכנס/לא נכנס מעולם/נצפה). בנוסף ניתן לצפות בכל delivery ע"י לחיצה עליו.
10. צפייה בדף delivery – מוצג מידע אודות הdelivery הנבחר. ישנם kbits – עם הסטטוס שנתן להם המשתמש (X, V, z), וכן ניתן לשנותם ע"י לחיצה בעמוד זה. כמו כן, מוצגים בעמוד זה גם הערות שרשם המשתמש לעצמו, הסרטון עצמו, וה-first level tree (הסבר מפורט לגבי first level tree ו-dependencies tree וההבדלים ביניהם מצורף בנספח ג). בנוסף, עבור לחיצה על כל איבר ב-first level tree ניתן לצפות ב-first level tree שלו.
11. אפשרויות שמירת flows – Past flows – שמירת flows שהתקבלו בערב ונצפו (גם אם נצפו חלקית בלבד).

- Wishlist flows – שמירת flows שהתקבלו בעבר, אך הלקוח טרם צפה בהם ומעוניין לצפות בהם בעתיד.
 - Personally defined flows – שמירת deliveries אחד אחד, באופן לא ממויין (בניגוד לשתי האפשרויות הקודמות, בהם הflow מגיע ממויין). כאן גם ישנה האפשרות לקבוע איזשהו סדר ולשנותו בהתאם לצורך.
 - Public flows – flows יכולים להיות מסומנים כ-public, כלומר שאר המשתמשים יכולים לצפות בהם ולא רק המשתמש ששמר/יצר אותם.
11. אינטגרציה ועבודה עם "מידע אמיתי" – כלומר אינטגרציה ותקשורת עם צד-השרת (הצוות השני).

3.3 מסמך דרישות הלקוח

Project: Web solution for viewing knowproj videos

Introduction

General

KnowProj is an experimental project to manage training/teaching content.

The current requirements document refers to a web application that enables a user to view training material supplied by the system, grade it, keep records of how he benefits from the training, and use this records to get more accurate training material on his next interaction with the system.

Source system overview (not part of this project)

The Source System's DB (not part of this part of the project) classifies the training content in a unique methodology, which was submitted in the past as provisional patent, but now permitted to use: The main content elements are the "delivery" and the "Kbits". The deliveries are short training lessons which is consumed by the learners (users of current project), and the Kbits are theoretical knowledge elements that can be learned when consuming the delivery. Thus, each delivery is linked to a list of Kbits that are taught in it. In addition, each delivery is accompanied with list of Kbits that are needed as prior knowledge, so that without them

the delivery can't be understood by the learner. Therefore a Kbit which is provided by one delivery can be the same one that is needed by another delivery, thus the dependency between the deliveries is created.

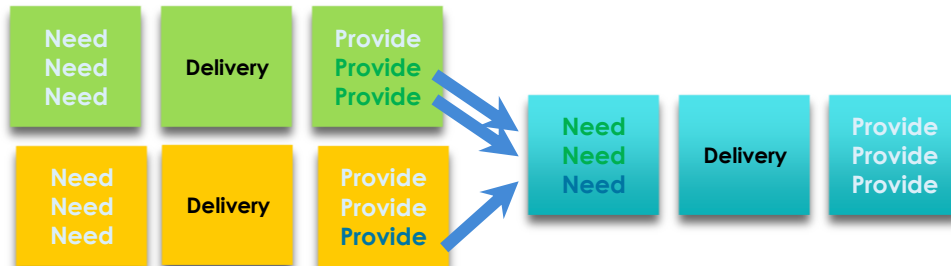


FIGURE 1 – TREE SAMPLE OF SOURCE-SYSTEM'S ELEMENTS

Apart from the deliveries and Kbits we have “terms”. Terms are short text parts that has meanings under certain scope of knowledge (this scope can be geographical, temporal, organizational, functional). Terms can be linked to deliveries or to Kbits. In cases where Kbits are explanation of a term, there's a strong link between them and they are almost interchangeable. However since it's not always the case, we still keep Kbits and terms as different entities.

Learner's system overview

The user start his interaction with the Learner's system by searching for a learning target. Once selected, the system provides him with a learning flow, which as the learning-target as the last element in the flow.

How does the system provide the initial learning flow?

The learner's system interact with the source-system (content-system) to get a full tree of dependencies based on the sample depicted in figure 1. A pre-defined algorithm in the learner's system will extract a learning-flow out out of the dependencies tree.

What can the user do with it?

The user can view the items (“deliveries”) in the flow one by one in order to get to the learning target. while doing so, he can view and grade the Kbits related to each delivery, thus provide feedback on what he truly knows and understand.

How does the system improve the learning flow?

After first usage, the system knows what the user knows, based on his grading of Kbits. This system uses this knowledge in the algorithm that selects the learning-flow, by skipping deliveries, rearranging them, or adding supporting deliveries.

Definitions

System (in the current project's scope) – the learner's system. the system which is used by the learner to gain knowledge.

User (in the current system's scope) – The Learner. The person who consumes training content. He/She does so by getting a “learning flow”, learning according to it, and provide personal feedback to better adjust future learning flows.

Learning flow – an ordered list

Delivery – the training lesson to be taught. Can be a video url.

KnowledgeBit or Kbits – a piece of knowledge that can be learned by the training lesson, or might be required to know before watching a training lesson

Element – a general term for delivery, term or knowledge_bit.

Functional requirements

General Work flow and client requirements

- Start work flow by one of the below:
 - Search
 - Search in element name or in element description,
 - search w/wo term mapping
 - search w/wo RegExp seach enabled
 - Retrieve old path that started
 - Retrieve friend's path
- Based on selection from search:
 - Show up to 3 flow options
 - Sort by Fastest for user or by tightest links (algorithm for fastest or tightest links will be provided – not part of the project), based on parameter of “how far back I remember”
 - Prefer things the user has seen before or new things to see (by user selection)

- View flow:
 - gray out things the user know (based on KBs and/or videos seen)
- During work:
 - Can view deliveries one by one, mark KBs with “V” “X” or “?” (which are saved in the user DB, along with time of change)
 - For each delivery – show first level tree, show content itself (assume video only), show characteristics, show KBs, enable personal comments that can be saved and edited
 - For each element – show first level tree, from tree click on links to:
 - show first level tree or characteristics
 - for delivery: also – view delivery, or add to flow for myflows, or open new flow (define as new learning target)
- The application should have an option to personally define flow. This means that next to each delivery there should be an option “add to flow” and give a flow name. there will be “my flows” section where a user can see:
 - past flows,
 - wishlist flows,
 - and personal flows. the personal flows are editable, and can be marked as “public” so that other users can find it. personal flow is a simple ordered list of deliveries, regardless of KBs.
- Session: must keep user's session and history, so that user can get back to where he was if the window closes. Also, there should be on option to get to the same point in a work-flow through URL.

Server2server:

Get tree recursive just for kb->delivery

Get tree single depth (first level) for any links

Search in element name, search in element description, search w/ term mapping – all with RegExp search enabled

Learner should be able to search based on things he saw/didn't see (“search ‘pattern’ only in videos I watched before”), and search in his own comments

Non functional requirement: Keep **objects** in JavaScript

Non Functional requirements

Performance requirements

Quality requirements

- Need to define set of tests to run on each version
- Need to define system demo
- Well documented with comments in code

Hardware/Software requirements

- We must use commercial apache server, have server-side in PHP, client side in JS, use AJAX for client requests from server through a set of defined APIs. DB is MySQL.
- The system should run on Firefox, Chrome, Safari and IE browsers.

Design constrains

- None of the known architectural pattern is required, but loose coupling is absolutely required so that new developers can focus on set of functionalities separately, with well-defined APIs between them.
- To enable each team to get a progress, “dummy requests” and “dummy results” need to be provided as first step
- ContentDB and usersDB should be kept separated. User's IDs from usersDB can be used in the ContentDB, and element's IDs can be used in the usersDB.

Management constrains

- 300 man hours is required per team member.
- Progress/demo will be presented once a month.
- Before starting the work there's a need to estimating hours needed per feature/requirement and prioritization for defining planned set of features by end of the project. Two months later we should go over the list again and refine it to have final agreement of set of features.
- First stage will be creating the interface between server and client, have “dummy requests” and “dummy responses” so that server and client can work independently from that moment onwards.

3.4. הגדרת המשימות

	Task description
1	platform work Create
2	Dummy requests and results between teams
3	Get input from user-learner
4	Create the request and send it to main server
5	Get answer from main server
6	Translate answer into learn flows
7	Sort and choose up to 3 learn flows
8	Show learner the suggested learn flows
9	Enable watching deliveries one by one
10	Enable marking kbts
11	Show deliveries data (for each delivery)
12	Show element's data (for each element)
13	Add flows options: past, wishlist and personally defined
14	Work with data supplied by main server (other teams)
15	Work with real content
16	Define system demo

4. תיאור התהליכים המרכזיים והאלגוריתמים

4.1. אלגוריתם מציאת flow אופטימלי מתוך עץ התלויות

When a user searches for a flow, we are sending his request to the main-server, and we get back as a response a dependency tree.

From this tree we have to extract an optimal flow.

The following algorithm describes the algorithm used in order to do it.

This algorithm has 2 parts:

Algorithm 1 - to get list of deliveries (greedy)

Have function for "detect minimum prereqs for delivery X", and work recursively
For each needed kbit – search all Deliveries that provide it. For each Delivery – run "detect minimum prereqs". You get something like this:

Kbit1	D1 – 4*	D2 - 5	D3 - 2	
Kbit2	D1 - 4			
Kbit3	D4 - 2	D5 - 3		

*it means delivery D1 requires minimum 4 deliveries as prerequisites

You have to search all combinations that provide together Kbit 1, 2, 3 – and see what the minimum is. The solution here is [D1-4], [D4-2] that gives total of 8 prereqs (4+2+D1 itself + D4 itself).

At the end you get a list of Deliveries. Remove duplicated Deliveries from the list to get the minimum list.

Algorithm 2 - to get ORDERED list of deliveries (greedy)

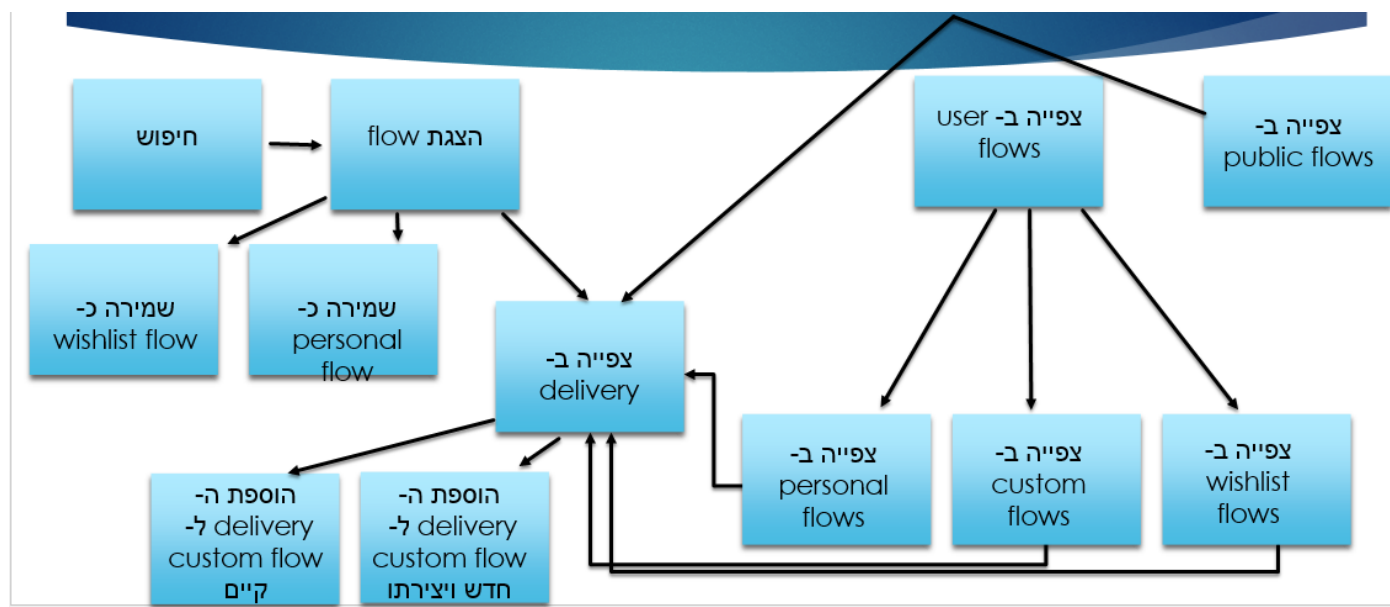
Function "order deliveries by precedence for target delivery X":

Start from the X and search down the tree (but skip deliveries that are already in the ordered list) until you get a node that has no children.

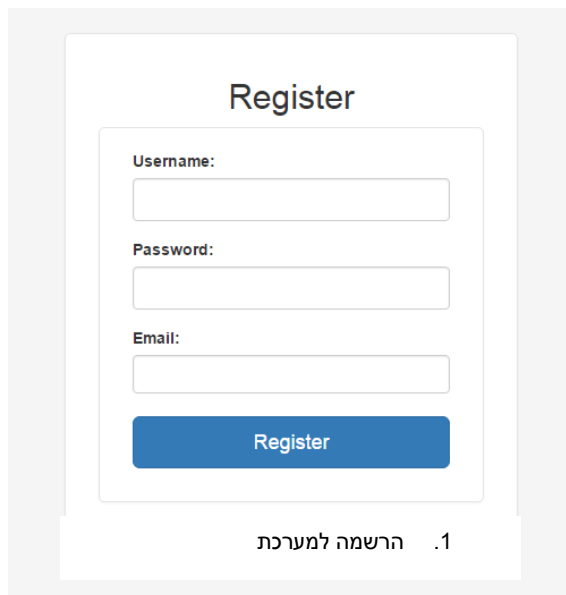
Add the node to the ordered list.

Go back to its parent and send it to "order deliveries by precedence..."

4.2. תהליכים מרכזיים



5. צילומי מסך עיקריים של המערכת



Register

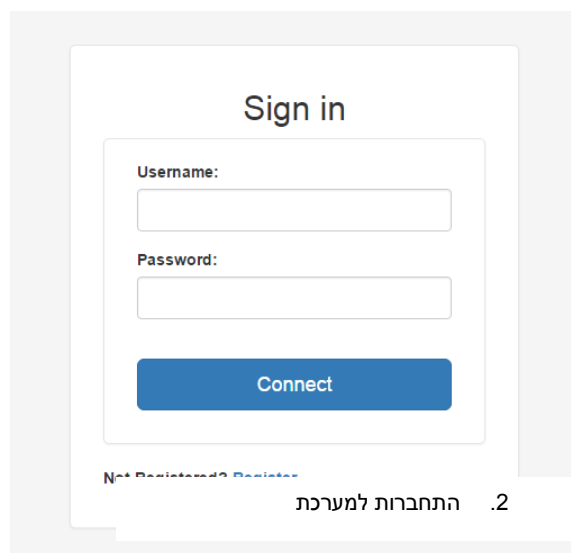
Username:

Password:

Email:

Register

1. הרשמה למערכת



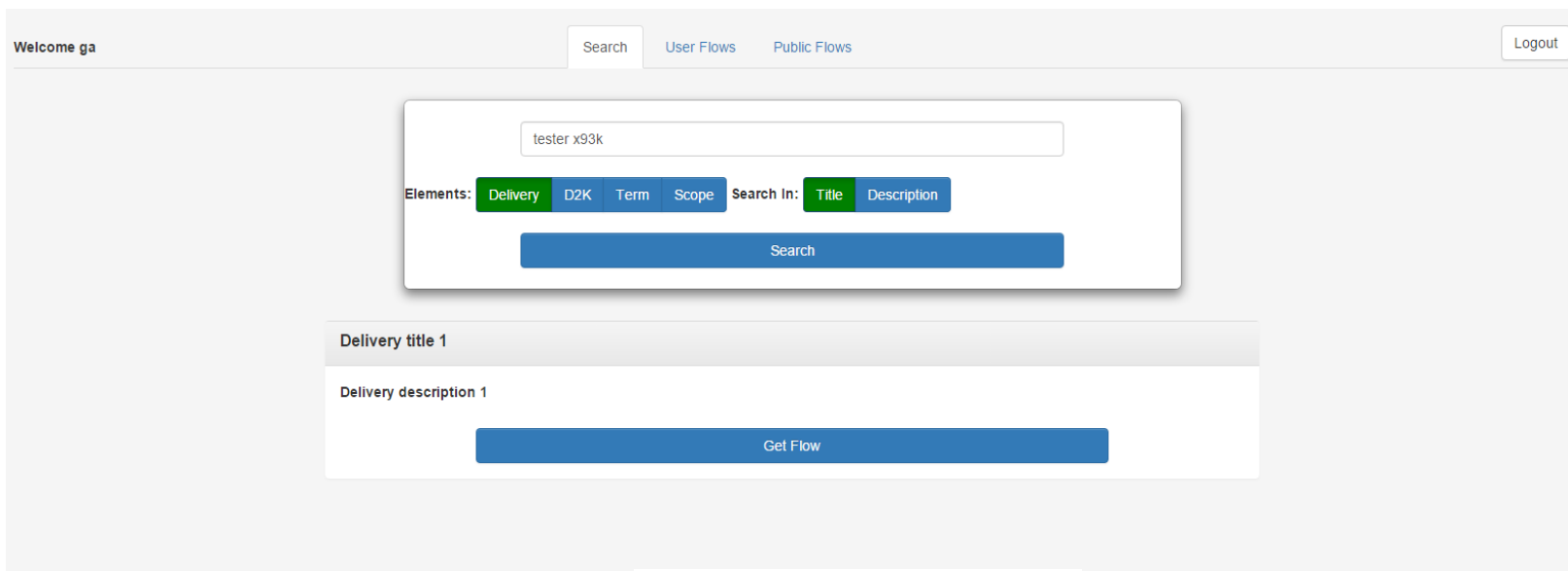
Sign in

Username:

Password:

Connect

2. התחברות למערכת



Welcome ga Search User Flows Public Flows Logout

tester x93k

Elements: **Delivery** D2K Term Scope Search In: **Title** Description

Search

Delivery title 1

Delivery description 1

Get Flow

3. חיפוש, וקבלת ה-flow

Welcome ga

Search User Flows Public Flows Logout

Add to Personal Flows Add to Wishlist

d07

d06

d02

d03

4. הצגת ה-flow (באמצעות deliveries וחיצים המסודרים עפ"י סדר הצפייה).

Welcome ga

Search User Flows Public Flows Logout

Add to Personal Flows Add to Wishlist

d07

d06

d02

d03

d01

d06

Description : d06_description

Kbits_needed:

k01 (V) (X) (2)

k02 (V) (X) (2)

k03 (V) (X) (2)

k04 (V) (X) (2)

Kbits_provided:

k07 (V) (X) (2)

Add To Custom Flow

Personal comments:

comments

Submit Comment

Deliveries_related: d02,d03

Terms: t02,t03

Scope: s01,s02

First Level Tree:

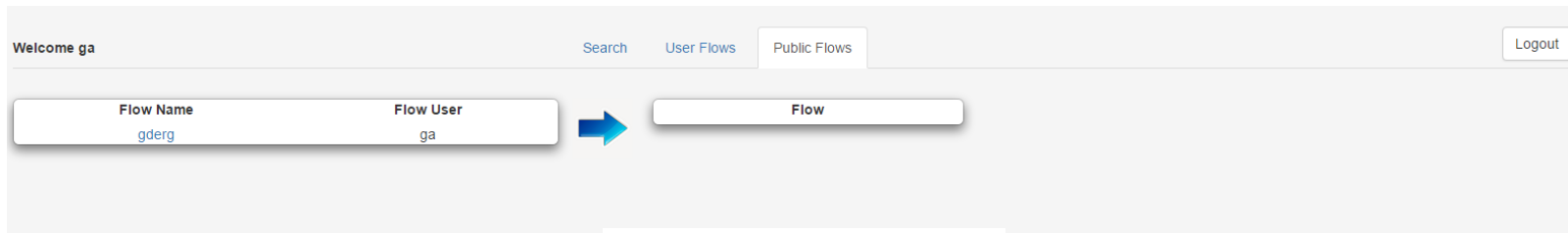
Entity First Level Tree:

שטיות שבוחרות אומרות במסעדות

5. עמוד הצפייה ב-delivery.



6. צפייה ב-user flows.



7. צפייה ב-public flows.

6. מסמכי בדיקות

1. בדיקת התחברות

- א. מטרת הבדיקה: לבדוק את מנגנון ההתחברות לאתר
 ב. כלי עזר: גישה למערכת הDB (PHPMYADMIN).
 ג. מהלך הבדיקה:

צעד	פעולה	תגובה צפויה
1	הכנס לאתר לכתובת index.php.	מופיעים שני שדות username ו- password וכפתור התחברות.
2	נסה ללחוץ על כפתור ההתחברות להתחברות כאשר השדות ריקות.	מופיעה הודעת שגיאה שהשדות ריקים.
3	הכנס נתונים לא נכונים ולחץ על כפתור ההתחברות.	מופיעה הודעת שגיאה שהנתונים לא נכונים.
4	הכנס נתונים נכונים שמופיעים בDB.	התחבר ועובר ישר לעמוד החיפוש.

2. בדיקת התנתקות

- א. מטרת הבדיקה: לבדוק את מנגנון ההתנתקות מהאתר
 ב. כלי עזר: אין
 ג. מהלך הבדיקה:

צעד	פעולה	תגובה צפויה
1	התחבר לאתר.	נמצא בחלון החיפוש.
2	לחץ על כפתור ההתנתקות.	עבר לחלון ההתחברות.
3	נסה להכנס לעמודים הבאים: Search.php , flows.php, Publicflows.php, Personalflows.php	עבור כל אחד מהעמודים זה חוזר לעמוד ההתחברות.

3. בדיקת חיפוש

- א. מטרת הבדיקה: לבדוק שמנגנון החיפוש בונה את הפקודה הנכונה.

ב. כלי עזר: הדפסות עבור

ג. מהלך הבדיקה:

צעד	פעולה	תגובה צפויה
1	בצע חיפוש ריק	הודעת שגיאה
2	בצע חיפוש עם הכיתוב 1 title אבל בלי סימון בelements	הודעת שגיאה
3	בצע חיפוש עם הכיתוב 1 title וסימון של delivery ובלי סימון search in	הודעת שגיאה
4	עבור כל אפשרות בצע חיפוש	הודעת שגיאה
5	הוסף הדפסה של מחרוזת ה json לפני שהיא נשלחת לאתר, כאשר הכל מסומן ועם הכיתוב 1 title.	<pre>{"searchtext":"title 1","elements":{"delivery":"true","d2k":"true","term":"true","scope":"true"},"field":{"title":"true","description":"true"}}</pre>

4. שמירת delivery נבחר

א. מטרת הבדיקה: לבדוק שמנגנון בחירת ה delivery שומר נכון את ה ID שצריך.

ב. כלי עזר: ערכו את הקובץ SetIDInSession.php שיחזיר את ID כתשובה.

ג. מהלך הבדיקה:

צעד	פעולה	תגובה צפויה
1	בצע חיפוש.	הופיעו תוצאות.
2	הדפסו את כל ה ID שקיבלנו בתוצאות חיפוש.	כתבו אותם בצד.
3	בחרו delivery.	הופיע כפתור getflow.
4	לחצו על הכפתור.	הופיעה הדפסה עם id.
5	בדוק את המקום שהופיע ה ID בפעם הראשונה ואיזה תוצאת חיפוש בחרת	אותו דבר.

5. הפקת פלואו

- א. מטרת הבדיקה: לבדוק שבטעינת העמוד flows.php חוזר פלואו.
 ב. כלי עזר: אין
 ג. מהלך הבדיקה:

צעד	פעולה	תגובה צפויה
1	בצע חיפוש ובחרו את אחת האפשרויות.	עובר לעמוד flows.php.
2	הדפיסו את הפלואו שהתקבל מהשרת.	הופיעה הדפסה.
3	הדפיסו את כל ה deliveries שהתקבלו.	הופיעה הדפסה . הפלואו התקבל בצד שמאל של החלון.

6. קבלת נתונים עבור כל delivery שהתקבל ב flow .

- א. מטרת הבדיקה: לקבל את כל הנתונים עבור כל delivery שיש לנו ב flow על מנת להציג את הנתונים.
 ב. כלי עזר: אין
 ג. מהלך הבדיקה:

צעד	פעולה	תגובה צפויה
1	בצע חיפוש ובחרו את אחת האפשרויות.	עובר לעמוד flows.php.
2	הדפיסו את הבקשה.	{"flow":["d07","d06","d02","d03","d01"]}
3	הדפיסו את התשובה (מצורפת תשובה רק עבור 1 delivery).	{"name":"d01","type":"d","description":"d01_description", "url":"http://www.youtube.com/embed/f-aBSV-rVXM", "kbits_needed":"k01,k02,k03", "kbits_provided":"k04,k07", "deliveries_related":"d02,d03", "terms":"t01,t02,t03","scope":"s01,s02","parent":null}

7. הוספת ה flow ל Personal Flows

- א. מטרת הבדיקה: לבדוק את המנגנון שמוסיף את ה flow לרשימת flows של המשתמש.
 ב. כלי עזר: phpmyadmin .
 ג. מהלך הבדיקה:

צעד	פעולה	תגובה צפויה
1	בצע חיפוש ובחרו את אחת האפשרויות.	עובר לעמוד flows.php.
2	לחץ על Add to Personal flows	הופיע שדה Flow Name וכפתור Save .
3	לחץ פעם נוספת על Add to Personal flows	נעלם שדה ה Flow Name וכפתור Save.
4	לחץ פעם נוספת על Add to Personal flows	הופיע שדה Flow Name וכפתור Save .
5	כתוב שם ל Flow ולחץ על כפתור Saven .	קפצה הודעת Added.
6	בדוק תחת User Flows בקטגוריה Personal Flow את שם ה Flows	הופיעה ה Flow .
7	לחץ על שם ה Flow ובדוק אם ה Flow באותו סדר	מופיע באותו סדר.
8	בדוק ב DB בעזרת PHPMYADMIN אם ה Flow מופיע בסדר הנכון (בטבלה PersonalFlows).	מופיע באותו סדר.

8. הוספת ה flow ל Wishlist Flows

- א. מטרת הבדיקה: לבדוק את המנגנון שמוסיף את ה flow לרשימת Wishlist של המשתמש.
- ב. כלי עזר: phpmyadmin .
- ג. מהלך הבדיקה:

צעד	פעולה	תגובה צפויה
1	בצע חיפוש ובחרו את אחת האפשרויות.	עובר לעמוד flows.php.
2	לחץ על Add to Wishlist flows	הופיע שדה Flow Name וכפתור Save .
3	לחץ פעם נוספת על Add to Wishlist flows	נעלם שדה ה Flow Name וכפתור Save.
4	לחץ פעם נוספת על Add to Wishlist flows	הופיע שדה Flow Name וכפתור Save .

5	כתוב שם לFlow ולחץ על כפתור Saven .	קפצה הודעת Added.
6	בדוק תחת User Flows בקטגוריה Wishlist Flow את שם הFlows	הופיעה הFlow .
7	לחץ על שם הFlow ובדוק אם ה Flow באותו סדר	מופיע באותו סדר.
8	בדוק בDB בעזרת PHPMYADMIN אם הFlow מופיע בסדר הנכון (בטבלה WishlistFlows).	מופיע באותו סדר.

9. הוספת ה Custom Flow לDelivery

- א. מטרת הבדיקה: לבדוק את המנגנון שמוסיף Delivery חדש Custom Flow ויוצר Custom flow חדש.
- ב. כלי עזר: phpmyadmin .
- ג. מהלך הבדיקה:

צעד	פעולה	תגובה צפויה
1	בצע חיפוש ובחרו את אחת האפשרויות.	עובר לעמוד flows.php.
2	לחץ על Add to Custom flows	הופיע שדה User Flows וכפתור Save .
3	לחץ על Add to Custom flows	נעלם שדה User Flows וכפתור Save .
4	לחץ על Add to Custom flows	הופיע שדה User Flows וכפתור Save .
5	בחר New Flow וכתוב שם לFlow .	קפצה הודעת Added.
6	בדוק בDB בעזרת PHPMYADMIN אם הFlow מופיע בסדר הנכון (בטבלה CustomFlows).	הופיעה שם ה Flow עם ה Delivery הנכון ו 0 Chunk .
7	בחר Delivery אחר והוסף אותו לרשימה החדשה.	קצפה הודעת Added .
8	בחר את אותו Delivery והוסף אותו לרשימה החדשה.	קפצה הודעת Exist .

10. הוספת הערות משתמש ל-Delivery .

- א. מטרת הבדיקה: לבדוק את המנגנון שמאפשר לשמור לכל delivery הערות של המשתמש ולערוך אותם בהמשך.
- ב. כלי עזר: phpmyadmin .
- ג. מהלך הבדיקה:

צעד	פעולה	תגובה צפויה
1	בצע חיפוש ובחרו את אחת האפשרויות.	עובר לעמוד flows.php.
2	בחר Delivery וכתוב לו הערה, לחץ על כפתור ה Submit Comment .	הודעה: Comment Updated .
3	בדוק בDB תחת הטבלה <u>userdetailsperdelivery</u> אם ההערה נשמרה.	ההערה נמצאת בDB תחת ה Delivery הנכון.
4	עדכן את ההערה ובדוק אותה שוב .	ההערה עודכנה בDB .

11. הוספת \עדכון ל Kbit Status .

- א. מטרת הבדיקה: לבדוק את המנגנון שמאפשר להוסיף או לעדכן מצב של ה KBIT .
- ב. כלי עזר: phpmyadmin .
- ג. מהלך הבדיקה:

צעד	פעולה	תגובה צפויה
1	בצע חיפוש ובחרו את אחת האפשרויות.	עובר לעמוד flows.php.
2	בחר Kbit וסמן לו את האפשרות V	V הופך לירוק.
3	בדוק בDB בטבלה <u>userskbits</u> תחת העמודה status	מעודכן ל V .
4	עדכן את ה KBIT ל X	X הופך לאדום ו V בלי צבע.
5	בדוק בDB בטבלה <u>userskbits</u> תחת העמודה status	מעודכן ל X .
6	עדכן את ה KBIT ל ?	? צהוב, X ו-V בלי צבע.

7	בדוק בDB בטבלה <u>userskbits</u> תחת העמודה status	מעודכן ל ? .
---	--	--------------

12. הצגת כל ה Flows של המשתמש .

- א. מטרת הבדיקה: לבדוק את המנגנון שמאפשר להציג את כל ה Personal Flows של המשתמש.
 ב. כלי עזר: phpmyadmin .
 ג. מהלך הבדיקה:

צעד	פעולה	תגובה צפויה
1	לחץ על User Flows.	נפתח בצד שמאל 3 קטגוריות Personal Flows , Custom Flows , Wishlist Flows .
2	לחץ על Personal Flows .	הופיעה רשימה של כל שמות ה Flow של המשתמש הנוכחי.
3	בחרה Flow כלשהו ולחץ עליו	הופיע ה Flow .
4	בדוק בDB בעזרת PHPMYADMIN אם אלו כל ה Flows של המשתמש הנוכחי ושכל אחד מציג את המידע הנכון.	מופיע בצורה תקינה.

13. הצגת כל ה Flows שהמשתמש שמר לעצמו.

- א. מטרת הבדיקה: לבדוק את המנגנון שמאפשר להציג את כל ה Wishlist Flows של המשתמש.
 ב. כלי עזר: phpmyadmin .
 ג. מהלך הבדיקה:

צעד	פעולה	תגובה צפויה
1	לחץ על User Flows.	נפתח בצד שמאל 3 קטגוריות Personal Flows , Custom Flows , Wishlist Flows .
2	לחץ על Wishlist Flows .	הופיעה רשימה של כל שמות ה Flow שהמשתמש שמר לעצמו.
3	בחרה Flow כלשהו ולחץ עליו .	הופיע ה Flow .
4	בדוק בDB בעזרת PHPMYADMIN אם אלו כל ה Flows של המשתמש	מופיע בצורה תקינה.

	שמר לעצמו ושכל אחד מציג את המידע הנכון.	
--	---	--

14. הצגת כל ה Custom Flows .

- א. מטרת הבדיקה: לבדוק את המנגנון שמאפשר להציג את כל ה Custom Flows של המשתמש.
- ב. כלי עזר: phpmyadmin .
- ג. מהלך הבדיקה:

צעד	פעולה	תגובה צפויה
1	לחץ על User Flows.	נפתח בצד שמאל 3 קטגוריות Personal Flows , Custom Flows , Wishlist Flows .
2	לחץ על Custom Flows Flows .	הופיעה רשימה של כל שמות ה Flow שהמשתמש יצר.
3	בחרה Flow כלשהו ולחץ עליו .	הופיע ה Flow .
4	בדוק בDB בעזרת PHPMYADMIN אם אלו כל ה Flows של המשתמש יצר לעצמו ושכל אחד מציג את המידע הנכון.	מופיע בצורה תקינה.

15. שינוי ה Custom Flow .

- א. מטרת הבדיקה: לבדוק את המנגנון שמאפשר לערוך את כל ה Custom Flows של המשתמש.
- ב. כלי עזר: phpmyadmin .
- ג. מהלך הבדיקה:

צעד	פעולה	תגובה צפויה
1	לחץ על User Flows.	נפתח בצד שמאל 3 קטגוריות Personal Flows , Custom Flows , Wishlist Flows .
2	לחץ על Custom Flows Flows .	הופיעה רשימה של כל שמות ה Flow שהמשתמש יצר.
3	בחרה Flow כלשהו ולחץ עליו .	הופיע ה Flow .
4	בחר Delivery האחרון מתוך ה flow ולחץ על החץ למעלה	Delivery עלה.

5	בדוק ב DB שה Delivery עלה.	ה Delivery עלה.
6	בחר Delivery הראשון מתוך ה flow ולחץ על החץ למטה	ה Delivery ירד.
7	בדוק ב DB שה Delivery ירד.	ה Delivery ירד.
8	בחר Delivery מאמצע ה flow ולחץ על כפתור המחיקה	ה Delivery נמחק.
9	בדוק ב DB שה Delivery נמחק וכל ה Deliveries מתחתיו עודכנו גם.	ה Delivery נמחק והרשימה עודכנה.
10	בחר Flow שהוא מוגדר כ Public Deliveries ומחק לו את כל ה Deliveries	כל ה Deliveries נמחקו פרט לאחרון שנתן את השגיאה , can't delete public flow , Please UnPublic it.
11	הורד את ה Flow ממצב Public ומחק את ה Delivery האחרון.	ה Delivery נמחק וכך גם ה Flow.
12	בדוק ב DB שה Flow נמחק.	ה Flow נמחק.

16. הוספת Flow ל Public .

- א. מטרת הבדיקה: להוסיף Custom Flow כ Public Flows .
- ב. כלי עזר: phpmyadmin .
- ג. מהלך הבדיקה:

צעד	פעולה	תגובה צפויה
1	לחץ על User Flows.	נפתח בצד שמאל 3 קטגוריות Personal Flows , Custom Flows , Wishlist Flows .
2	לחץ על Custom Flows	הופיעה רשימה של כל שמות ה Flow שהמשתמש יצר.
3	לחץ על כפתור ה Set as Public	הכיתוב השתנה ל Set as UnPublic
4	בדוק ב DB אם ה Flow הפך להיות Public	ה Flow במצב Public .
5	בדוק בעמוד ה Public Flows אם נמצא ה Flow הספציפי	נמצא.

6	לחץ על כפתור ה Set as UnPublic	הכיתוב השתנה ל Set as Public
7	בדוק ב DB אם ה Flow הפך להיות UnPublic	ה Flow במצב UnPublic .
8	בדוק בעמוד ה Public Flows אם נמצא ה Flow הספציפי	לא נמצא.

17. הצגת Public Flows .

- א. מטרת הבדיקה: הצגת כל ה Flows שמוגדרים כ Public .
- ב. כלי עזר: phpmyadmin .
- ג. מהלך הבדיקה:

צעד	פעולה	תגובה צפויה
1	וודא שאין Flows שמוגדרים כ Public Flows .	
2	לחץ על Public Flows	הודעה: No Public Flows!
3	הוסף Custom Flow כ Public	
4	לחץ על Public Flows	ה Flow מוצג בטבלה.
5	הוסף Custom Flow ל Public עם התחברות כמשתמש אחר	
6	לחץ על Public Flows	ה Flow מוצג בטבלה עם המשתמש החדש.

7. סיכונים טכניים וקשיים למימוש דרישות הלקוח והדרך להתגבר עליהם

1. בעית אינטגרציה עם הצוות האחראי על צד השרת

הפרויקט שלנו, כפי שתואר במסמכי הגדרת הפרויקט מורכב משלושה חלקים:

- א. צד שרת
- ב. צד לקוח
- ג. צד הלומד

הצד שלנו, צד הלומד אמור לתקשר עם צד השרת ולקבל ממנו נתונים עבור מספר קריאות (למשל: קבלת נתונים אודות Delivery מסויים, קבלת תוצאות חיפוש של משתמש).

בתחילת הדרך, הגדרנו כמשימה – לעבוד עם "Dummy data" – כלומר, לבנות לעצמנו stubs, שמדמים לנו את התשובות שנקבל מצד השרת.

היו לנו מספר ישיבות עם צוות צד השרת וכן עם מנחה הפרויקט בניסיון לבנות איזשהו מבנה אשר יהיה מקובל על שני הצדדים, שיכיל את התשובות / הבקשות עבור שני הצדדים, ואשר בעזרתו כל צד יבנה את ה"Dummy data" שלו.

כבר בטפסי הדיווח הראשונים דיווחנו על בעית התנהלות עם צוות השרת, אשר היינו צריכים "לרדוף" אחריהם לצרכי קבלת מבנה הנתונים, ולאחר מחשבה, החלטנו ליצור בעצמנו את מבנה הנתונים ולשלוח להם אותו.

לקראת סוף הפרויקט, כשכבר עמדנו לפני שלב האינטגרציה, קבענו ביוזמתנו, ולאחר תיאום עם מנחה הפרויקט, פגישת אינטגרציה עם הצוות השני – על מנת להתחיל לעבוד עם נתונים אמיתיים ולהגיע לאיזשהי מערכת כללית עובדת.

יום לפני פגישה האינטגרציה, הודיע הצוות האחר כי אינו יכול להיפגש (מאילוצי מערכת – נזקקו לשרת אמיתי על מנת לבצע את האינטגרציה, אשר לא סופק להם ע"י מנחה הפרויקט).

פגישת האינטגרציה בסופו של דבר נקבעה לתחילת חודש ספטמבר (קרוב מאוד למועד ההגשה), ובה הבחנו כי צוות השרת לא יצר (בחלק מהמקומות) את מבנה הנתונים (אשר נשלח כתשובה לשרת שלנו) באופן אשר הוסכם בתחילת הדרך!

בכל זאת, ביצענו מאמצים להתאים ולשנות את מבנה הנתונים אצלנו כך שיתאים למערכת שלנו, והצלחנו לעשות זאת, אך גילינו כי השרת עובד בצורה מאוד איטית אשר לא מאפשרת לנו הצגה של כל עבודתנו כראוי.

דרך ההתמודדות – החלטנו להשאיר את המערכת שבנינו עם stubs עבור התקשורת אל מול צד השרת, וזאת על מנת לא לפגוע בתוצרים ובכל מה שעבדנו עליו במהלך השנה. עם זאת – יצרנו מחדש stubs אשר תואמים את מבנה הנתונים הנשלח מצד השרת, וזאת על מנת שפרויקטי עתיד יוכלו לאחד את הפרויקטים ביתר קלות.

*השאירו את קוד האינטגרציה כבלוקי הערות בתוך קבצי הקוד אשר מהווים את stubs.

2. אופן שמירת ה-Flows בטבלאות במסד הנתונים

Flow מוגדר כרצף בעל אורך מסויים של Deliveries.

הבעיה מתחילה בכך שהאורך אינו ידוע מראש, ויכול להיות ארוך ככל שנרצה.

לכן, קיימות בעיות באופן הגדרת העמודה המייצגת את רשימת ה-Deliveries בטבלת ה-Flows, כך שנוכל לדעת שהיא מכילה בוודאות את כל ה-Flows האפשריים, שכן, כפי שנאמר קודם, אין אורך מקסימלי הידוע מראש.

דרכים אפשריות לפתרון:

ניקה לדוגמא את ה-Flow הבא:



א. ניצור טבלה המכילה את העמודות: flow_id, data, chunk_number.

נגדיר אורך קבוע לעמודה data, וכך נוכל לשים כמות מוגדרת מראש של deliveries_names בעמודה זו – אך עם זאת, נוכל ליצור מספר גדול כרצוננו של שורות עבור אותו flow_id, עם deliveries_names נוספים, ואת הסדר של השורות נוכל לציין בעזרת העמודה chunk_number. המפתח הראשי של הטבלה יהיה מורכבת מ-3 השדות, על מנת למנוע כפילויות.

איך תראה הטבלה עבור ה-flow הנ"ל:

(נניח כי הגדרנו שהאורך של העמודה data יכול להכיל לכל היותר 3 שמות של deliveries)

Flow_id	data	chunk_number
001	D01,D04,D07	0
001	D03,D08,D19	1

ב. אפשרות שנייה, ניצור בדיוק את אותה טבלה, רק שהפעם לא נשתמש ממש ב-chunks של delivery_names, אלא בעמודה האמצעית יהיה לנו delivery_name אחד בלבד, ולא כמה ביחד.

הטבלה הפעם תראה כך:

Flow_id	data	chunk_number
001	D01	0
001	D04	1
001	D07	2
001	D03	3
001	D08	4
001	D19	5

*הערה – בטבלאות האמיתיות קיימת גם עמודת "Users", אך אין בה צורך עבור ההסבר.

התלבטנו כיצד כדאי לממש את הטבלה.

היתרון בטבלה הראשונה – יותר "נעים לעין" עבור בן אדם שרואה את הטבלה עצמה לנגד עיניו – מספר מצומצם של שורות, וכך יותר קל לו להבין מה משמעות הטבלה וכיצד היא בנויה.

מצד שני, הטבלה השניה בנויה באופן נוח יותר לשימוש – החיפוש קצר וקל יותר, אין צורך לעבוד עם מחרוזות ארוכות.

מנסיוננו כמתכנתים, בחרנו באופציה השניה, שכן היא נוחה בהרבה בתהליך העבודה, וכן נכונה יותר מבחינת עקרונות התכנות.

8. עמידה ושינויים בדרישות לקוח

8.1. עדכוני לו"ז של תהליך הפרויקט ועמידה בדרישות הלקוח

	Task description	Finish date	status
1	platform work Create	1/30/2015	✓
2	Dummy requests and results between teams	2/9/2015	✓
3	Get input from user-learner	2/23/2015	✓
4	Create the request and send it to main server	3/10/2015	✓
5	Get answer from main server	3/18/2015	✓
6	Translate answer into learn flows	4/20/2015	✓
7	Sort and choose up to 3 learn flows	5/5/2015	* ✓
8	Show learner the suggested learn flows	5/19/2015	✓
9	Enable watching deliveries one by one	6/9/2015	✓
10	Enable marking kbts	6/22/2015	✓
11	Show deliveries data (for each delivery)	7/8/2015	✓
12	Show element's data (for each element)	7/24/2015	✓
13	Add flows options: past, wishlist and personally defined	8/3/2015	✓
14	Work with data supplied by main server (other teams)	8/7/2015	✓

15	Work with real content	8/14/2015	* ✓
16	Define system demo	9/1/2015	✓
17	Add option – public flow	**	✓
18	Show and enable editing user message for each delivery	**	✓
19	Show FLT – First Level Tree for each delivery	**	✓
20	Show 2 nd FLT – for each entity in the 1 st FLT (except for deliveries) – show it's FLT	**	✓
21	Enable editing (only by the owner user) the personally defined flows	**	✓

8.2 שינויים בדרישות הלקוח

סעיפים אשר הותאמו מחדש לדרישות הלקוח:

1. Sort and choose up to 3 learn flows – בעקבות מצוקת זמנים והתעכבויות שונות שנבעו בעיקר מנסיונות שיתוף פעולה ויישור קו עם הצוותים האחרים בפרויקט, הוחלט כי הדרישה תשתנה להצגת flow אחד בלבד, ובפרויקטי המשך יעשו את הנדרש ויגדילו את כמות הflows בהתאם לנדרש. הבסיס והאלגוריתם, שהם כמובן החלק היותר משמעותי, מומשו על-ידינו.
2. Work with real content – כפי שיתואר בהמשך ספר הפרויקט, בפרק "סיכונים טכניים וקשיים למימוש דרישות הלקוח", הייתה בעית אינטגרציה, ובסופו של הדבר הפרויקטים נשארו מנותקים והבקשות ממומשות ע"י stubs. גם זאת, באישור ובהנחיית המנחה. פרטים נוספים ניתן לקרוא בפרק הנ"ל.

סעיפים הנ"ל נוספו עם התקדמות הפרויקט והבנת צרכי המערכת. **

9. סיכום דוחות חודשיים ומפגשים

9.1. סיכום הדוחות החודשיים והפגישות עם הלקוח

מועד הפגישה	סיכום הפגישה	
דצמבר	מפגש היכרות עם הפרויקט, הבנת מטרותיו והרעיונות העומדים מאחורי יישומו	1
דצמבר	הגדרת הפרויקט ושלבי העבודה הנדרשים	2
דצמבר	פגישת היכרות של 3 הצוותים השותפים לפרויקט. דיון בנוגע לתשתית התקשורת בין חלקי הפרויקט	3
פברואר	מפגש עם המנחה ו-3 הצוותים. סיכום תהליך התקשורת בין 3 הפרויקטים. בנוסף הגדרנו מטלות למימוש עד לפגישה הבאה	4
מרץ	דיון בנוגע ל-flows, עצי התלויות, אופן מימושם ושמירתם במערכת	5
אפריל	דנו בנוגע למצב ההתקדמות, דיברנו על שאלות שעלו לנו במהלך העבודה עד כה, והכנו תוכנית עבודה לחודש הקרוב. מנחה הפרויקט הנחה אותנו והסביר לנו את ההבדלים בין סוגי העצים השונים, דבר שלא היה ברור לגמרי ממסמך דרישות הלקוח	6
מאי	עיקר המפגש היה העברת המידע לגבי האלגוריתם היוצר את flow האופטימלי מעץ התלויות שהתקבל מהשרת המרכזי, ע"י מנחה הפרויקט	7
יולי	פגישת סנכרון, בדיקת מצב התקדמות, הצגת דמו של המערכת עד כה, שאלות, תיקונים והתכווננות לסיום הפרויקט	8
ספטמבר	מפגש אינטגרציה בין 2 צוותים (מערכת הלומד – אנחנו, והשרת המרכזי) בו יצרנו התקשרות בין שני השרתים ועבדנו עם שרת חינמי הפועל באינטרנט.	9

10. סיכום הפרויקט ומסקנות

העבודה על הפרויקט הייתה מאתגרת ומעניינת מאוד.

זו אינה הפעם הראשונה שאנו לוקחים פרויקט ומתחילים אותו מאפס, אך התנסינו בכלים חדשים שלא הכרנו קודם לכן (wamp server, Eclipse), וכן בשפות תכנות חדשות עבורנו (JS, php, html5).

העבודה עם מנחה מהתעשייה הייתה מאוד מרעננת ושונה מכל הפרויקטים שעשינו עד היום בתואר, ובהחלט נהנינו במהלך העבודה.

תוצאות הפרויקט הינן טובות מאוד, שכן המערכת עובדת ומספקת את כל הפונקציונליות הנדרשת, וניתן להרחיבה בקלות.

עם זאת, האינטגרציה בסופו של דבר בוטלה עקב קשיים טכניים (שהוצגו בפרק 7), אך הפרויקט הותאם לאינטגרציה עתידית, ומפתחי התוכנה העתידיים יוכלו בקלות להתאים את המערכת לעבודה עם שרת חיצוני.

נשמח לראות שבעתיד המערכת אכן תעבוד בחברת אינטל ותשמש את עובדיה.

ברצוננו להודות למנחה הפרויקט, מר יניב לביא ולד"ר ענת אהרוני על ההנחיה והליווי במהלך השנה ועל הענקת הכלים הדרושים להצלחת הפרויקט.

ביבליוגרפיה

<http://www.w3schools.com>

<http://www.wampserver.com/en>

<https://eclipse.org/downloads>

<https://angularjs.org>

נספחים