**Revision: 11**

**Revision history**

Describe the changes made and the reason for the change between the revisions.

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Date** | **Description of changes** | **Author** |
| P1.0-1 | 2016-05-31 | Created draft version | Hariharan |
| P1.0-2 | 2016-06-09 | Refined after the meeting with Solution owner Tim & other stake holders | Hariharan |
| P1.0-3 | 2016-06-20 | Updated the NFR requirements | Hariharan |
| P1.0-4 | 2016-06-22 | Incorporated some of the review comments from Rahul. | Hariharan |
| P1.0-5 | 2016-07-15 | Updated solution outline section | Sivarajan Chitraikani |
| P1.0-6 | 2016-07-22 | Reviewed and updated | Abhijit Guhagarkar |
| P1.0-7 | 2016-08-08 | Reviewed and add comments | Nha-Tien Nguyen |
| P1.0-8 | 2016-08-15 | Updated the document based on Nha-Tien review comments | Sivarajan Chitraikani |
| P1.0-9 | 2016-08-23 | Updated the document based on Tim Ellmers review comments | Sivarajan Chitraikani |
| P1.0-10 | 2016-08-29 | Review done by IKEA PMs | Tim Ellmers, Rolf Medina |
| P1.0-11 | 2016-08-31 | Updated the document based on IKEA PMs & Nha-Tien review comments | Sivarajan Chitraikani |
| P1.0-12 | 2016-09-02 | Final comments and changes from IKEA | Nha-Tien Nguyen |

|  |  |
| --- | --- |
| **Reviewed/Approved** | **Name** |
| Yes/No |  |

|  |  |
| --- | --- |
| **Approved deviations from Solution Design** |  |
| After clicking 'view jobs' or at the link on the Job Title in the Job Results list, candidates will navigate to a Module Pop up that is giving a pre-view of the Job Description (Main source of this description is the ATS taken in via API into the Microsite Database. Detailed job search results shall be shown in Kenexa Brassring within the IKEA webpage. | 2.2 i |
| Only “View Jobs” will be displayed as action button at the job result list. All other Kenexa Action Buttons will not be displayed at this point. It can be find at the ATS page. | 2.2.8  5.1  5.1.3.1 8 |

## Abbreviation

|  |  |
| --- | --- |
| Abbreviation | Explanation |
| ATS | Applicant Tracking System |
| AWS | Amazon Web Services |
| CDN | Content Delivery Network |
| DAO | Data Access Object |
| DNS | Domain Name System |
| IRW (FS) | IKEA Retail Web ( |
| IRW (MW) | IKEA Retail Web (Mobile Web) |
| JST | Job Search Tool |
| KB | Kenexa Brassring |
| NWP | New Web Platform |

Table of Contents

[Abbreviation 3](#_Toc460580537)

[1 Introduction 5](#_Toc460580538)

[2 Business current state / Solution 6](#_Toc460580539)

[2.1 Capability description 6](#_Toc460580540)

[2.2 Feature description 7](#_Toc460580541)

[2.3 References 10](#_Toc460580542)

[2.4 Performance & Response Time 10](#_Toc460580543)

[2.5 Tech Stack 11](#_Toc460580544)

[3 Landing to Microsite: 12](#_Toc460580545)

[4 Content management Tools for Multilingual data 13](#_Toc460580546)

[5 Solution Outline 14](#_Toc460580547)

[5.1 High Level Solution Design: 14](#_Toc460580548)

[5.1.1 Akamai CDN 14](#_Toc460580549)

[5.1.2 Amazon Cloud 15](#_Toc460580550)

[5.1.3 Components Built by Capgemini 15](#_Toc460580551)

[5.1.4 JST Batch Frequency 19](#_Toc460580552)

[5.2 Open Source Libraries 19](#_Toc460580553)

[5.3 Software Requirements 19](#_Toc460580554)

[5.4 Hardware Requirements 20](#_Toc460580555)

[5.5 Business Requirements 20](#_Toc460580556)

[6 Platform rules 21](#_Toc460580557)

[6.1.1 IRW.RULE.TRACEABLE 21](#_Toc460580558)

[6.1.2 IRW.RULE.ENABLEDISABLE 21](#_Toc460580559)

[7 Non functional requirements 22](#_Toc460580560)

[7.1 Responsive web design 22](#_Toc460580561)

[7.2 Adaptability 22](#_Toc460580562)

[7.3 Maintainability 22](#_Toc460580563)

[7.4 Scalability 22](#_Toc460580564)

[7.5 Accessibility 22](#_Toc460580565)

[7.6 Performance Considerations 23](#_Toc460580566)

[7.7 Browser Support 23](#_Toc460580567)

[7.8 Security Considerations 23](#_Toc460580568)

[7.9 SEO 23](#_Toc460580569)

[7.9.1 Implementations for SEO 23](#_Toc460580570)

[7.10 Web Analytics 23](#_Toc460580571)

[7.11 Back up and Restoration 23](#_Toc460580572)

[8 Out of Scope 25](#_Toc460580573)

# Introduction

The objective of this solution is to redesign the current job search tool and develop one global solution that works seamlessly on the following platforms.

1. IRW (FS)
2. IRW (MW)
3. New Web Platform (NWP)
4. IKEA Inside
5. ICO Worker.

This document describes the solution design approach of the Job Search tool.

# Business current state / Solution

Today in IRW the IKEA Jobs search page uses a custom solution to present dropdown options for searching all IKEA jobs, globally. The current functionality is difficult to maintain or change.



* Sub Issue 1: Current job search selection functionality is not available for internal candidates (IKEA Inside).
* Sub issue 2: Lack of interface between Kenexa and Job Search module. High manual maintenance effort on IRW, NWP and IBM.
* Sub issue 3: Job search functionality has language as first selection criteria and that needs to be changed to country as first selection criteria to allow a satisfying candidate experience. Moreover a suggestive search result in dropdown lists and a display of search results are requested enhancements.
* Sub issue 4: Not all countries are connected to the Job Search Tool on IKEA.com (for example, Russia) due to failure to integrate newly created Russian TG with a Job Search Tool.
* Sub issue 5: The current job search tool does not allow to switch on the WCAG technology for the responsive design candidate experience within Kenexa Brassring. This is preventing solution enhancement.

## Capability description

User Stories

Business Rules

Workflows

Capability

**Customer Benefits**

* Reduced Efforts – No efforts required on individual platforms. As the solution is generic, changes implement in microsite will reflect in all platforms.
* Better availability and usability of job search functionality for internal and external candidates.
* Easy maintenance of Job Search Tool values using API technology from ATS to the Job Search tool.
* Accessibility via mobile devices without losing core functionality.
* Better candidate experience.
* Better control on Look & Feel - Since we are extracting the Kenexa response and we have the flexibility to control on Look & Feel.

## Feature description

The global solution will implement the following features:

1. Redesign how these dropdown values are served and update the IKEA Jobs search to use a Brassring API instead of a custom IRW or NWP solution.
2. Update all the dropdown values to be fetched from Brassring Excel data.
   1. Change the display & order of the search criteria. ie display the filter criteria of the search (candidate can always change the filter criteria) (Example from Benteler careerpage)
   2. Real time automatic synchronization of jobs and near time of all search criteria (language, location, org unit, etc.) between ATS and new Job Search tool.
   3. Possibility to limit search filter values depending on a previous filter selected (for example, when a country is selected, availability of values in other filters would be limited to options for that country)
   4. Possibility to have specified filters available on a Job Search Tool and in the following order of display:
      * Country
      * Location
      * Work area
      * Job type
      * Language
   5. Implement keyword search.
   6. All filters can be selected separately without following an order.
   7. Display number of found jobs per selection once a criteria was selected.
   8. Reset filter in different sub categories
   9. After clicking 'view jobs' candidates will be directed into Kenexa Brassring. Source tracking is needed.
   10. Advance search option will be provided in microsite. Click on advance search link will open Kenexa Advance search page. This is a Kenexa solution and will not be easily adaptable with other ATS.
   11. Possibility to have a suggestive search (when a candidate starts typing a combination of letters possible suggestions appear) in filter search like Country, Location, Work area, Job type and Language drop downs.
   12. Possibility to get results in the keyword field regardless the language of query
   13. Key word search and filter search will work together and keyword search results can be filtered using the filter criteria selected by the user.
   14. Filter criteria in filter search is displayed and can be delete by using the ‘X’ in the criteria
   15. Pagination: 10 jobs are displayed per page and pagination is displayed at right top and right bottom of the job result list.
3. The country ID in microsite url decides if Advance Search and Log in is shown. For Russia both links will not be displayed following the legal needs in Russia not allowing routes into Kenexa without asking for the nationality.
4. The language IDs in microsite url decide on the language of the Microsite. This language ID will be the decision point what language is displayed.
5. List of countries and languages supported are provided below

|  |  |  |  |
| --- | --- | --- | --- |
| Market Name | Language | Language Code | Country Code |
| Belgium | ***Dutch (NL)*** | nl | BE |
| Belgium | ***French (FR)*** | fr | BE |
| Czech Republic | ***Czech*** | cs | CZ |
| Denmark | ***Danish*** | da | DK |
| Germany | ***German*** | de | DE |
| Austria | ***German*** | de | AT |
| Switzerland | ***German*** | de | CH |
| Switzerland | ***French (FR)*** | fr | CH |
| Switzerland | ***Italian (IT)*** | it | CH |
| Italy | ***Italian (IT)*** | it | IT |
| Spain | ***Spanish (ES)*** | es | ES |
| France | ***French (FR)*** | fr | FR |
| Croatia | ***Croatian*** | hr | HR |
| Hungary | ***Hungarian*** | hu | HU |
| Netherland | ***Dutch (NL)*** | nl | NL |
| Norway | ***Norwegian*** | no | NO |
| Poland | ***Polish*** | pl | PL |
| Portugal | ***Portuguese*** | pt | PT |
| Romania | ***Romanian*** | ro | RO |
| Russia | ***Russian*** | ru | RU |
| Slovakia | ***Slovak*** | sk | SK |
| Serbia | ***Serbian*** | sr | RS |
| Finland | ***Finnish*** | fi | FI |
| Sweden | ***Swedish(SV)*** | sv | SE |
| United States | ***English*** | en | US |
| Australia | ***English*** | en | AU |
| Ireland | ***English*** | en | IE |
| United Kingdom | ***English*** | en | GB |
| Canada | ***English*** | en | CA |
| Canada | ***French (FR)*** | fr | CA |
| China | ***English*** | en | CN |
| China | ***Chinese*** | zh | CN |
| India | ***English*** | en | IN |
| Japan | ***English*** | en | JP |
| Japan | ***Japanese*** | ja | JP |
| Malaysia | ***English*** | en | MY |
| Malaysia | ***Malaysian*** | ms | MY |
| Singapore | ***English*** | en | SG |
| South Korea | ***English*** | en | KR |
| South Korea | ***Korean*** | ko | KR |
| Thailand | ***English*** | en | TH |
| Thailand | ***Thai*** | th | TH |

1. New header & footer will be developed based on IKEA style and guidelines.
2. Russia will have an own JST microsite web design to hide Kenexa links “Advanced Search” and “Log in” from people entering IKEA.RU.
3. Russia will have an own JST microsite webdesign to hide Kenexa Action buttons “Save to basket” and “Create Search Agent” from people entering IKEA.RU.
4. All Talent Gateways will stay as is. When the Full Job Description is viewed the candidate is already on the Gateway that is controlled by Kenexa. For Russian Jobs this also means the Avnet redirection will kick in.
5. Keyword search is able to search within the full Microsite Database, including Job title, Job Description, Country, Location, Work Area, Language
6. The Microsite Data Base includes Job Title, Job Description, Country, Location, Work Area, Language and Job type information. All Infos are taken via APIs calling Kenexa Brasssring on non-Russian and Russian gateway.
7. The JST Microsite is differentiating between Internal and External Usage and has different Gateways connected to the Datebases
8. By clicking on ‘Back to Jobsearch’ the job search results are still displayed (no need to re-enter the job search criterias).
9. Pop up for job description for ‘Close’ and ‘Continue’ and navigate back and forth.
10. At the job result list all the jobs displayed can be select by one click and de-select again
11. Several jobs can be selected and the job description can be viewed (via the pop up modal) and navigate back and forth to see job description

## References

[1] BSGF\_IKEA Jobs search tool Requirements Doc –

<https://service.projectplace.com/pp/pp.cgi/0/1231561036?direct=1231410099#/tab_>

[2] IBM Kenexa API Reference -

<https://service.projectplace.com/pp/pp.cgi/0/1197196474?direct=1231569121#/tab_>

[3]. Site Extension guideline reference

<https://service.projectplace.com/pp/pp.cgi/0/1231563340?direct=1253275997#/tab_>

## Performance & Response Time

Response time is from end user perspective. Performance tests will be performed using load runner. Average response times for various actions and maximum number of users that Job Search Tool can handle will be benchmarked.

Expecting Response Time values for the below table from Business:

|  |  |
| --- | --- |
| Use Case | Response Time in Milli Seconds |
| Search page loading with all dropdown values |  |
| Update Total Job count in search page when criteria changes |  |
| Search Jobs without any filter criteria. |  |
| Search Jobs with Multiple filter criteria. |  |
| Keyword search. |  |
|  |  |
| Clicking Next page in Job Results. |  |
|  |  |

## Tech Stack

* + Java – 1.8 Version
  + Front Tier – Java script, Jquery, JSP, HTML & CSS. Ajax calls will be used for populating drop downs.
  + Integration Tier – Spring framework 4.0, Spring MVC, Spring IOC, Kenexa web service will be consumed in Spring Batch Programs.
  + Data Access Tier – Spring JDBC, Brassring Kenexa API.
  + MySQL DB
  + IDE – Eclipse Mars 4.5.0.
  + Code Repository – Clear case
  + Development Web server – Apache Tomcat 8
  + Production Server – Tomcat on Linux server
  + Build tool - Jenkins

# Landing to Microsite:

All the platform’s current Job search tool link will be replaced with new microsite url. Parameters will be passed from source platform to microsite to identify the source platform & locale in the microsite. Based on the parameters passed, microsite will display the page in the appropriate language.

secure.ikea.com requires PCI DSS compliance. JST doesn’t process/store any data which requires PCI DSS compliance. Later we can migrate to www.ikea.com from ww8.ikea.com, when https is enabled in ikea.com domain. AKAMAI manages the [www.ikea.com](http://www.ikea.com) domain. There is no target date available for enabling https in www.ikea.com.

Microsite URL format for external

[https://ww8.ikea.com/ext/job/search/external/[COUNTRY\_CODE]/[LANG\_CODE]/home](https://ww8.ikea.com/ext/job/search/external/%5bCOUNTRY_CODE%5d/%5bLANG_CODE%5d/home)

Microsite URL format for IKEA Inside Jobs (internal)

[https://ww8.ikea.com/ext/job/search/internal/[COUNTRY\_CODE]/[LANG\_CODE]/home](https://ww8.ikea.com/ext/job/search/internal/%5bCOUNTRY_CODE%5d/%5bLANG_CODE%5d/home)

The following two parameters has to be passed from source platform

COUNTRY\_CODE – ISO code for country

LANG\_CODE – ISO code for language

For INSIDE/ ICOWORKER there is no Country of Language ID, so the language and gateways need to be selected before navigation to the microsite happens. Current solution in INSIDE would be a possible way to have more links for different Gateways.

# Content management Tools for Multilingual data

There won’t be a need to add to have a content management system handling multilingual data. Instead internationalization property file will be implemented. All translated data will be available in a property file, from where it will read and written to the JSP. Internationalization property files will have key value pairs. Keys are used in Job Search Tool as reference and values contains language specific text those needs to be shown to user. Internationalization property files will be stored in server file system and that will be referenced by Job Search Application. Clear case will be used for version control of properties files.

Excel sheet containing the translated texts are placed in <https://service.projectplace.com/pp/pp.cgi/0/1267349738>. Property files will be generated based on excel sheet placed in this folder.

Property files have to be copied to the file system of Server containing the production JST Application.

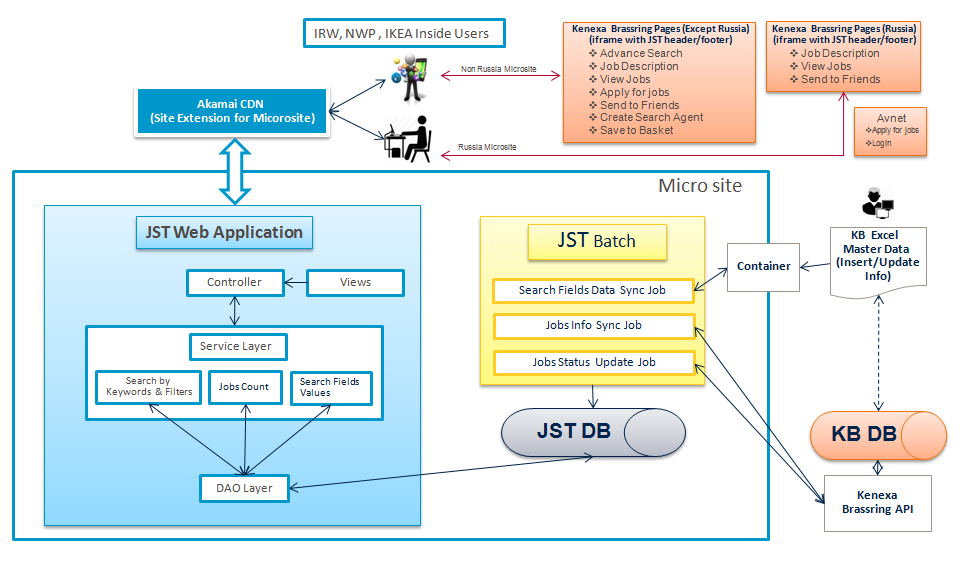


Example of the excel list that feeds the Property File.

# Solution Outline

The existing Job Search tool will be redesigned in to new global solution. The new global solution will have its own header, banner, footer, menu as per Ikea Look and feel guidelines. The back end Kenexa Brassring api will be reused as it is. Current Job search tool will be replaced with the url of this global solution.

## High Level Solution Design:



JST – Job Search Tool

KB – Kenexa Brassring

DAO Layer – Data Access Object Layer used for database integration

KB DB – Kenexa Brassring Database

JST DB – Job Search Tool Database

Container – Amazon S3 will be used as containerController – Handles the user requests using Service Layers

Views – Display information

Service Layer – Takes care of requests from controller using DAO Layer

Note: Apply for Jobs, Send to Friends, Create Search Agent and Save to Basket are out of scope for first release due to limitation in integrating with Kenexa Brassring Site

### Akamai CDN

Akamai CDN will provide DNS service for linking ww-8.ikea.com domain with Amazon Web Services Hosting. Akamai is not used for hosting any pages and caching.

### Amazon Cloud

The following products from Amazon are used for hosting Job Search Tool Microsite

AWS Beanstalk – Tomcat 8.0 (JST Web Application)

AWS RDS – MySql 5.7

AWS S3 (Container) – Brassring Excel Files hosting

AWS Beanstalk (Worker Environment) – JST Batch Application

The user access to AWS environments are controlled by Amazon CDT Team.

### Components Built by Capgemini

JST Web Application component is built by Capgemini to provide response UI for Job Search Page and to link the existing IRW, NWP and IKEA Inside pages

JST Batch Application component is built to provide the Job Synchronization using Brassring API and Search Fields Values Synchronization.

JST DB is used for storing Job information and Search filters dropdown values

#### JST Web Application

The following functionalities are covered as part of JST Web application implementations

1. Header and Footer in JST Pages
   1. IKEA specific Header and Footer will be displayed in home page, job description page, login page and advance search page
2. Home Page
   1. Home page will contain text box for keyword and filter search.
   2. Dropdown values for search filter fields are populated from JST DB using DAO Layer
   3. User can select the search filters and enter keyword together
   4. On click of Search button, search results will be shown
   5. Search result will be displayed in the home page
   6. Search result can be sorted and pagination is available
   7. Pagination and Sorting will be performed using Service and DAO Layer (Server side)
   8. On click of Reset button, keyword textbox will be emptied and selected fields will be reset
   9. Search can be performed using keyword and predefined fields like Country, Location, Work Area, Job Type & Language together. Jobs Information in JOB\_INFO & JOB\_INFO\_DESC tables of JST Database will be used for searching. DAO Layer will be used by Service layer for performing combined keyword search
   10. JST DB: column ‘SYSTEM\_TYPE ‘in JOB\_INFO table will be used for differentiating the external and internal jobs
   11. JST DB: column ‘COUNTRY ‘ in JOB\_INFO table will be used for differentiating the country specific jobs (for differentiating Global TG jobs and Russian TG jobs)
   12. Link to ‘Working at the IKEA Group’ and the employer branding picture
   13. Text for ‘How to apply’ section
3. Job Description Modal Popup (Job Title Click)
   1. On click of Job title, Job description modal popup will be displayed
   2. User will have option to close or continue in modal popup
   3. On click of continue, job description page will be loaded (iframe to Kenexa)
   4. On click of close, modal popup will be closed
4. Tips Popup for search
   1. On click of tips link, user will be tips content in modal popup
   2. Tips content will be language specific
5. Job Description Page
   1. Job description page will be created using iframe
   2. Brassring job description url is the source for iframe
6. Login Page
   1. Login page will be created using iframe
   2. Brassring job login url is the source for iframe
7. Advance Search Page
   1. Advance search page will be created using iframe
   2. Brassring job advance search url is the source for iframe
8. Jobs Count Information
   1. Total jobs count is shown on loading of homepage
   2. Jobs count will be updated whenever user makes selection in search filter
   3. Jobs count will be updated when user completes the keyword and focus is moved to other fields
9. View Jobs Modal Popup
   1. Total jobs count is shown on loading of homepage
10. Restrictions on keyword
    1. Special characters like ~()><@+- cannot be used as part of keyword
    2. The asterisk serves as the truncation (or wildcard) operator. Words match if they begin with the word preceding the \* operator

|  |
| --- |
| Keywords are case insensitive. Keyword Search will be performed using the logic provided belowScenario 1:  Keyword entered by User = *manager*  The jobs satisfying any one of the following conditions will be returned   * Job title with ***manager*** * Job description with ***manager*** * Country with ***manager*** * Location with ***manager*** * Language with ***manager*** * Work Area with ***manager*** * Job Type with ***manager***   Scenario 2:  Keyword entered by User = *manager Helsingborg Sweden*  The jobs satisfying any one of the following conditions will be returned   * Job title with ***manager*** *or* ***Helsingborg*** *or* ***Sweden*** * Job description with ***manager*** *or* ***Helsingborg*** *or* ***Sweden*** * Country with ***manager*** *or* ***Helsingborg*** *or* ***Sweden*** * Location with ***manager*** *or* ***Helsingborg*** *or* ***Sweden*** * Language with ***manager*** *or* ***Helsingborg*** *or* ***Sweden*** * Work Area with ***manager*** *or* ***Helsingborg*** *or* ***Sweden*** * job Type with ***manager*** *or* ***Helsingborg*** *or* ***Sweden***   Scenario 3:  A phrase that is enclosed within double quote (“"”) characters matches only rows that contain the phrase literally, as it was typed  For Example, Keyword entered by User = “system developer”. Jobs that contain the exact phrase “system developer” will be returned  Note :  Other filters (like Country, Location, WorkArea, Language, Job Type) will be applied on Jobs returned by Keyword Search Logic |

1. Implementation of Action Buttons like Apply Job(s), View Job(s), Send to Friends, Create Search Agent, Save to basket (Brassring UI will be linked using iframe).
2. Advance Job Search Page (Brassring UI will be linked using iframe)
3. Login Page (Brassring UI will be linked using iframe)
4. Russia users will not see Login and Advance Search Link in home page.
5. Html5 local storage will be used for storing the job search selections and job results. Major browsers to be supported by JST have html 5 storage support. Old browsers may not support html 5 so cookies will be used for storing the job search selections and ajax calls will be used for reloading the job result page, when the user clicks on “Back to Job Search” button.

#### JST Batch Application

The following functionalities are covered as part of JST Batch application implementations

1. Search Fields Data Sync Job - using the master data from Kenexa Brassring
   1. Amazon S3 will be used as container and excel files listed below needs to be uploaded
      1. BR\_CONFIG.xlsx contains the Brassring configuration information like url, site ids etc
      2. BR\_JST\_FIELDS\_VALUES.xlsx contains the values for search fields like country, location, workarea, job type and language
   2. Job will download BR\_CONFIG.xlsx and BR\_JST\_FIELDS\_VALUES.xlsx
   3. If the files are available in S3 then the following steps will be processed otherwise job execution will be stopped at this stage
   4. Information in BR\_CONFIG.xlsx will be used to populate/update JST\_CONFIG table
   5. Information in BR\_JST\_FIELDS\_VALUES.xlsx will be used to populate/update the below tables
      1. COUNTRY & COUNTRY\_VALUES
      2. LOCATION & LOCATION\_VALUES
      3. JOB\_TYPE & JOB\_TYPE\_VALUES
      4. LANGUAGE & LANGUAGE\_VALUES
      5. WORKAREA & WORKAREA\_VALUES
   6. After successful populate/update of data, the excel sheet will be renamed as BR\_CONFIG\_Processed\_{TimeStamp}.xlsx & BR\_JST\_FIELDS\_VALUES\_Processed\_{TimeStamp}.xlsx in S3 folder

Note: Only authorized people will have the right to upload files to Amazon S3 and it will be similar to uploading the file in website. User can do it using their browser.

Excel files are placed in <https://service.projectplace.com/pp/pp.cgi/0/1240910216> for reference

1. Jobs Info Sync Job - Jobs Info Caching using Kenexa Brassring API
   1. Brassring API will be called according to language list in JST\_CONFIG table information for Global and Russian TG. Only jobs posted after last update of information will be requested
   2. Site Id and locale site id specific to the language are specified in the Brassring API XML request
   3. DatePosted tag is added to Brassring API XML request
2. Jobs Status Update Job - Jobs information will be stored/updated in JOB\_INFO and Job description will be stored in JOB\_INFO\_DESC table. Jobs Info status update using Kenexa Brassring API
   1. Brassring API will be called according to language list in JST\_CONFIG table information for Global and Russian TG.
   2. Unposted Jobs will be marked as INACTIVE in STATUS column of JOB\_INFO table in JST database
   3. Site Id and locale site id specific to the language are specified in the Brassring API XML request
   4. <ReturnUnpostedJobs>Yes</ReturnUnpostedJobs> is added to Brassring API XML request

### JST Batch Frequency

The batches are scheduled from

a) KB Excel Data to Container and then JST Batch

b) KB API to JST Batch

|  |  |  |
| --- | --- | --- |
| Job Name | Schedule | Comments |
| downloadjobs | Every 1 hour at 00 min | Jobs download from Brassring: (b) KB API to JSt Batch |
| updatejobs | Every 1 hour at 30 min | Un posted Jobs Update:  (b) KB API to JSt Batch |
| syncfieldvalues | Every 1 hour at 45 min | Update the fields only master data is available in S3:  (a) KB Excel Data to Container and then JST Batch |

## Open Source Libraries

The following open source libraries will be used for developing Job Search Tool

1. Java Development Kit 1.8.0\_91
2. Spring MVC Framework 4.2.6.RELEASE
3. Spring Batch Framework 2.1.7.RELEASE
4. Apache POI 3.14

## Software Requirements

The following softwares are required for deploying Job Search Tool

1. Apache Http Server 2.4.23
2. Apache Tomcat 8.0.36
3. MySql Server 5.5

Note: 64 bits version of softwares will be used

## Hardware Requirements

Hardware used should have following specifications

1. RAM - 8GB
2. Hard Disk - 512 GB
3. Operating system - Linux/Windows with 64 bit OS

## Business Requirements

Business requirements Document (BRS\_for\_sign\_off.docx) is available in project place https://service.projectplace.com/pp/pp.cgi/r1231410099.

# Platform rules

*This section describes platform specific rules that should be considered in the design and implementation of the requirements. If IT-requirement exists for the solution, the Rules can be found in the IT-requirement.*

### IRW.RULE.TRACEABLE

Clear case will be used to version control (lifecycle) this change.

### IRW.RULE.ENABLEDISABLE

NA.

# Non functional requirements

## Responsive web design

Pages will be developed using HTML 5 and Bootstrap framework to support responsive web design. Pages developed in Microsite are aimed at allowing desktop webpages to be viewed in response to the size of the device one is viewing with.

## Adaptability

* Integration layer (Java layer) integrates with Kenexa API through request/response xml messages.  Based on front end form parameters, request xml is formed in Integration layer. The xml structure is defined by Kenexa.
* The interfaces will be designed in a more abstract / generic way that will allow for easy extension to other ATS. i.e. Integration tier is  kept loosely coupled with Kenexa which will give flexibility for adding any another ATS in the future.
* **Switching to different ATS** - Code changes required in Integration layer in order to form the request message as per New ATS tool standards. Code has to be written for downloading jobs from new ATS and integrated into JST Application. The amount of rework depends on new ATS’s request/response format i.e. XML or JSON or any other structure.

## Maintainability

Since all the platforms use a Global solution it will be easy to maintain. i.e. Changes implemented in Global solution will reflect in all platforms and a API call is sending over newest changes that has been made in locations or other fields.

The values for fields (country, Location, Work Area, Job Type, Language) are not coming from Brassring API so the values has to be inserted/updated using master files from Brassring Admin. Master files are present in <https://service.projectplace.com/pp/pp.cgi/0/1240910216> for reference.

When a new language comes into……

* language specific property file has to be created and placed in JST Application Server location
* BR\_CONFIG\_V1.xlsx file has to be updated with site id for new locale and uploaded to S3 folder for synchronization

## Scalability

Capacity of the servers will be increased based on the load to production systems. Architecture will support hosting the application in clusters

## Accessibility

Web Content Accessibility Guidelines (WCAG 2.0) is a stable, referenceable technical standard. It has 12 guidelines that are organized under 4 principles: perceivable, operable, understandable, and robust.

For each guideline, there are testable success criteria, which are at three levels: A, AA, and AAA. JST solution accessibility will be **‘AA’** compliant.

## Performance Considerations

Job information will be fetched from Brassring and stored in JST Database. This will be done using batch process separate from JST Web application. JST Web application will not invoke Brassring for fetching the data. This will improve the performance of JST application. JST database will act caching layer for job information.

## Browser Support

1. Global browsers with fast update pace. - Firefox 47.0 & 46.0 , Chrome 51.0.2704 & 50.0.2661
2. Global browsers with slow update pace. - Safari 5, 4 & 3, IE 11, 10 & 9 and Edge 25, 23 & 21.
3. Local/regional browsers those are very common in a specific market - Yandex 16 & 15. (Russia)

## Security Considerations

SSL enabled and within ikea.com domain. Solution will be tested for standard Ikea security tests such as SQL injection, Cross-site scripting & hacking.

## SEO

Solution follows the SEO guidelines such as Editable page title, metadata, URL, headings and page content should be part of new solution. Ikea DNS naming standards will be followed. To get better ranking in search engine, Microsite will be hosted with in ikea.com as sub domain using Site Extensions. For e.g. [ww8.ikea.com/ext/job/search](http://www.ikea.com/ext/job/search).

URL is not part of the SEO as url will not be language specific.

### Implementations for SEO

Added the following meta tags to support SEO

keywords

title

description

All contents displayed in jst pages are coming from language specific property file. So content is editable and search engine can index the content.

## Web Analytics

Standard web analytics measurements will be done ie Number of users visiting the Job Search page & Job results page will be tracked. Solutions will be developed as per the web analytics guidelines from Ikea.

## Back up and Restoration

JST Web, Batch are hosted inside Amazon EC2 Instance as Amazon Beanstalk Application and MySql is hosted inside Amazon RDS

Back up and restoration of JST Web, Batch and MySql Database will be implemented using the below AWS features.

Using AMI to Back Up EC2 Instances

AWS stores system images in what are called Amazon Machine Images (AMIs). These images consist of the template for the root volume required to launch an instance.

Use the AWS Management Console back up the root volume as an AMI.

Using Amazon RDS for Backups

Amazon RDS includes features for automating database backups. Amazon RDS creates a storage volume snapshot of database instance, backing up the entire DB instance, not just individual databases. Amazon RDS provides automated backups and DB snapshots for backing up and restoring DB instances

# Out of Scope

1. Removing the existing Job search tool from all platforms and linking to microsite is out of scope of this project. i.e removing the current page & updating the page with microsite url.
2. Any modifications to Kenexa API
3. Making Kenexa pages responsive.
4. Once the application navigates to Kenexa or an other
5. Apply for Jobs, Send to Friends, Create Search Agent and Save to Basket are out of scope for first release due to limitation in integrating with Kenexa Brassring Site