CandEx: Web Application for Candidate Engagement

B. Tech Project Report

By

Anushree Sharma (1310110066)

Rishabh Jain (1310110264)

Supervisors

Anirban Mondal
Assistant Professor, Computer Science and Engineering
anirban.mondal@snu.edu.in

Anandhu Dhanapal
Full Stack Developer, ViaSat India
anandhu.dhanapal@viasat.com

Vinod Sivadas
HR Head India, ViaSat India
vinod.sivadas@viasat.com

Submitted in the partial fulfillment of
requirements for B. Tech. in Computer Science
and Engineering

SHIV NADAR UNIVERSITY

Department of Computer Science and Engineering,
School of Engineering, Shiv Nadar University,
Gautam Buddha Nagar, U.P., India, 201314
http://www.snu.edu.in

Approval Sheet

This report titled CandEx by Anushree Sharma and Rishabh Jain is approved for the degree of B. Tech in Computer Science and Engineering.

Project Advisors

Name Anandhu Dhanapal

Signature ____________________

Date 24/04/2017

Name Vinod Sivadas

Signature FOR VS SN Swamy

Date April 24 2017

Name Anirban Mondal

Signature ____________________

Date ____________________

Declaration Sheet

We declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Name of the Student ANUSHREE SHARMA Signature

Date 24/04/2017

Name of the Student RISHABH JAIN Signature

Date 24/04/2017

Acknowledgement

We would like to express our sincere gratitude to Mr. Anandhu Dhanapal for his enormous support and guidance during the course of this entire project which helped us to successfully complete our project.
Additionally we would like to thank Mr. Vinod Sivadas for his constant advice and valuable feedback.

Above all, we are indebted to ViaSat, Inc. and its employees for giving us a chance to work in such a rich collaborative environment which allowed a free exchange of ideas and helped us to grow in holistic manner.

Lastly we would like to thank Dr. Anirban Mondal for his valuable suggestions and for taking keen interest in the progress of CandEx and our performance.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

HR	Human Resource
IBM	International Business Machines Corporation
UI	User Interface
DRF	Django RestFramework
SMTP	Simple Mail Transfer Protocol
HTML	Hyper Text Markup Language
MVC	Model View Controller
PHP	Hypertext Preprocessor
VCS	Version Control System
HTTP	Hypertext Transfer Protocol
FTP	File Transfer Protocol
CSS	Cascading Style Sheets
MIT	Massachusetts Institute of Technology
CTC	Cost to company

1

Introduction

1.1 Overview

CandEx is a tool, developed for the HR department at ViaSat Inc. to help them manage requisitions and candidate. It'll allow the recruiters to manage open positions, map suitable candidates to requisitions, handle a wide variety of candidate information, schedule interviews, manage candidate feedback and it automates the task of sending reminder mails to candidates or hiring managers. Additionally it also captures data relevant to the hiring process of a candidate and generates and displays reports in form of charts and data tables based on the same.

1.2 Objective

The goal of CandEx is to manage candidate personal and hiring process related information and further generate meaningful metrics which provide the HR team and the leadership actionable insights into the way the hiring process takes place at the company.
CandEx facilitates the process of scheduling interview rounds and recording the decision and feedback for each candidate. In addition to that it captures candidate feedback about the hiring process as well.
It allows the Hiring Manager to track the progress of open requisition of their interest and look at feedback for every candidate interviewed for the same.
It also automates the task of sending reminder emails for various stages of the candidate engagement process which takes place once a candidate is hired.

1.3 Motivation

Previously at ViaSat Inc., India the recruiters were using IBM's BrassRing tool to manage requisitions and record candidate data. BrassRing is more of a requisition oriented tool and does not provide it's users a way of capturing interview process related candidate information which the HR team was interested in. So in addition to this tool they use to maintain excel sheets that helped them capture other relevant fields related to the hiring process which became cumbersome as they were entering and managing data at two different places.

Furthermore the HR Team and Leadership are interested in looking at some metrics regarding the requisitions and the candidates that sit for the interview process for instance a diversity report of the candidates hired or the age of open requests, in order to achieve this data was taken from the excel sheets and charts and reports were prepared manually.

This gave rise to a need for an application that could enable its users to enter data at one place and capture all relevant details specific to the HR department's needs and something that could automate certain tasks like sending reminder mails for interview or preparing a bunch of reports every month and sending it across to the leadership. And this is how the idea of CandEx was introduced.

1.4 Scope

The scope of this project is to allow the users of the application to create, delete, update and manage new requisitions and candidates and to schedule interviews. It provides a platform for different interview panels of the same candidate to interact with each other and stores their comments and feedback for a candidate allowing them to reach to a common faster decision making. It enables the user to look at trends and patterns in the hiring process by exploiting the captured data and using it to generate reports, which lets the recruiting team to understand what changes they need to make to their hiring process to make more efficient thus giving them a valuable insight.

2

Related Work

IBM Kenexa BrassRing

IBM Kenexa BrassRing is a popular applicant tracking system which helps HR professionals to manage requisitions and find suitable candidates for their organization. This tool is used as a recruiting solution by roughly half of the Fortune 50 companies across various countries. It employs the mobile and social resources to find the most suitable candidates for a given requisition along with providing a personalized user experience.

BrassRing focuses more on finding the right talent for a company from a global database of candidates and is great at managing high applicant volume which makes the task of recruiters easier. So in that sense it is more requisition oriented.

BrassRing features:

- Flexible in terms of integration with other vendor systems.
- Automation of email message delivery
- Easy to use with an intuitive UI
- Highly scalable and fit for handling large amount of applicants
- Provides good candidate experience by centralizing all activities on one platform.
- Searches for candidates global which means bigger reach.

How is CandEx different from BrassRing?
IBM Kenexa BrassRing is a tool used widely across companies and countries. But CandEx is developed to cater to the specific needs of the HR team at ViaSat, Inc., India which means that it is customized to serve special and quite specific requirements. So CandEx was tailor-made to automate tasks and make the job of the HR department easier at ViaSat, Inc. India which was achieved by working very closely with the HR department there.CandEx is more candidate oriented capturing very specific fields relevant to a candidate’s hiring process.

3

Feasibility Study

Technical Feasibility

Technical feasibility study refers to whether project is technically feasible in terms of what technical requirements it serves, a project is said to be technically feasible if it is able to fulfill all the technical requirements.

CandEx uses DRF for backend and angular.io for frontend, in addition to that Git is used for version control and AWS for deployment.

There exists considerable amount of support in form well documented libraries for the technologies being used for us to achieve the functionalities expected from the application. However in Angular being a recently released web application platform poses some challenges as certain components are not very stable.

Economic Feasibility

Economic feasibility measures how financially possible it is to run the project and it also involves studying the return from the project basically measuring if the project is worth investing in or not.
CandEx uses DRF and Angular which are open source and free which makes the application highly feasible in this respect since there no costs involved in purchasing of license etc.

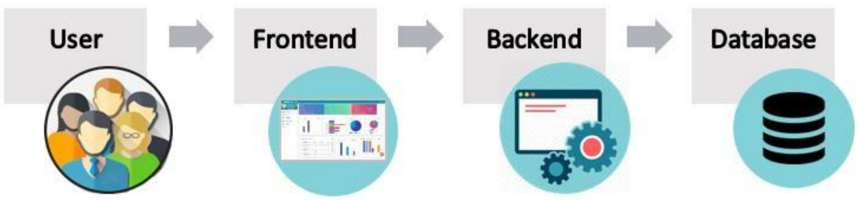
Operational Feasibility

CandEx provides an improved user experience which simple and hassle free. It is common platform for various tasks that the recruiters wished to perform. It uses two servers one of which is a lightweight server for frontend and another one is the backend server which makes it easy for the application to scale and it uses minimum number of third party dependencies.

4

Architecture

The representational diagram describing the architecture of the application is given below:



Frontend

The frontend is built with Angular.io which is a opensource typescript based framework for building client side applications while NodeJS is used to power up the angular application.

Gulp is a task builder that is responsible for loading node modules and uglifying JavaScript files for a faster and hassle free build of the client side application.
Materilize is a UI component that is used for beautifying the webpage. It is based on material design by Google Developers.
In an angular application there exists a component tree with a root component and its other child components which may have further child components and so on.

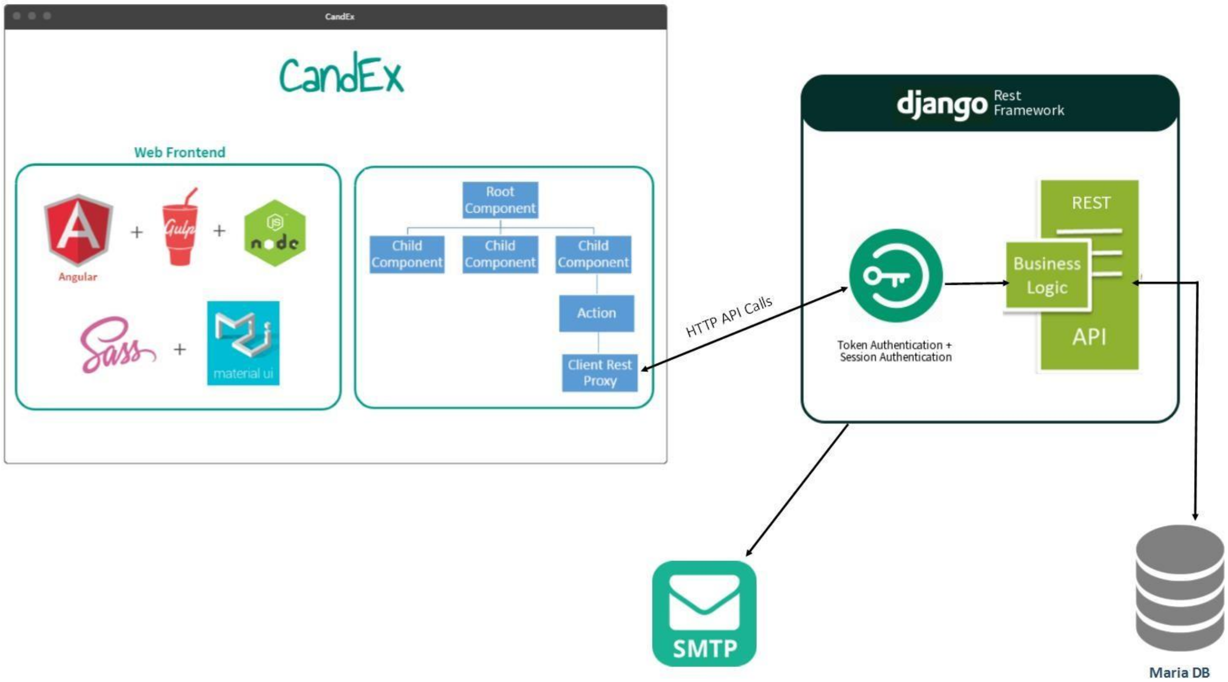
Backend

At the backend, Django Rest Framework (DRF) is used to build APIs that handle the business logic of the application. Token authentication at the backend for authenticating the frontend.SMTP that stands for Simple Mail Transfer Protocol is used to achieve the functionality of sending mails from the application.

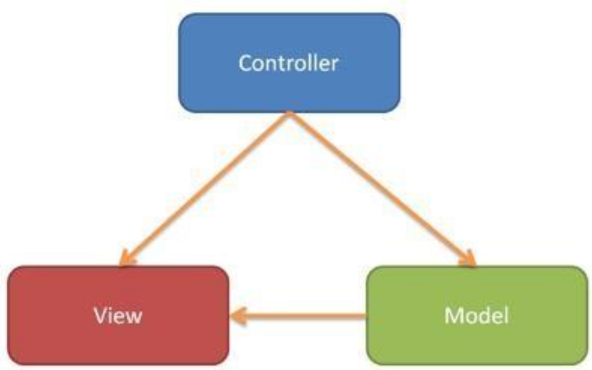
Database

MariaDB is used as the database management system which is a relational DB management which is built by the same set of people who have previously built MySql.

Before moving on to the technologies it is vital to discuss MVC architecture on which both Angular.io and Django RestFramework are based on.



MVC Architectural Pattern

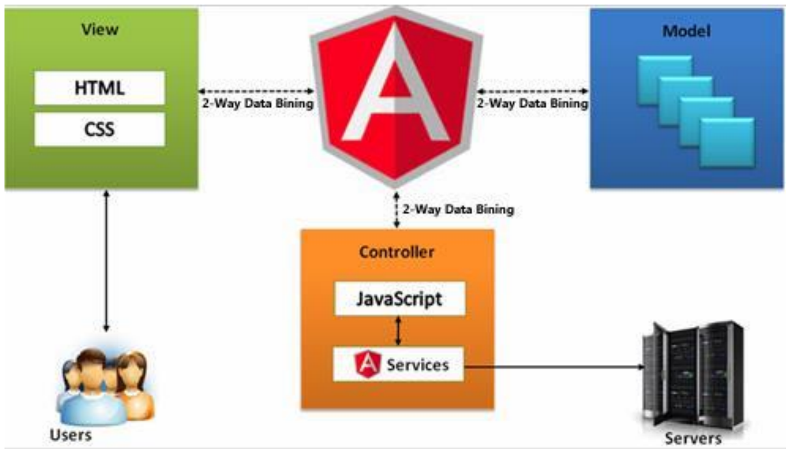


MVC is an architectural pattern that separates an application into three main logical components:

- Model – This component corresponds to all the data related logic that the user works with. This can represent the data that is being transferred between the view and controller.
- View – This component is used for all the UI Logic of the application. View includes all the UI Components that the final user interacts with.

- Controller – Controller act as an interface between model and view components to process all the business logic , manipulate the data using the MODEL component and interacts with the VIEWS to render the final output.

MVC in Angular:

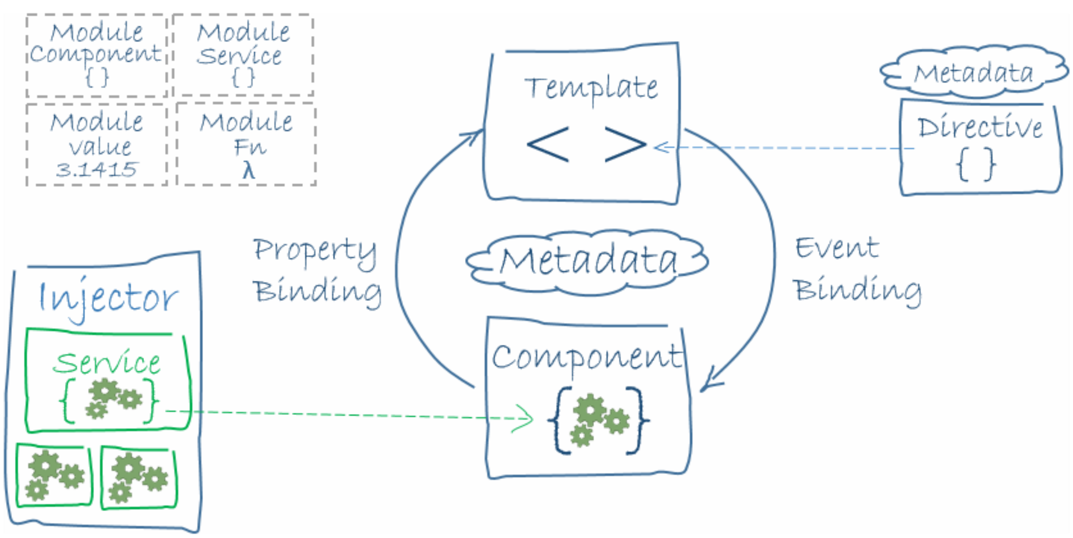


5

Technologies Used

Angular.io
Angular,io or Angular 2, simply Angular is a frontend web framework which is open source and based on typescript developed by the same team that built AngularJS to build client applications in HTML and JavaScript. It is a MVC framework for building applications in browser that serve complex requirements.

An angular application is made of components. They are built by composition of HTML templates which are managed in component classes and the corresponding application logic is written in services while modules handle the boxing of components and services.



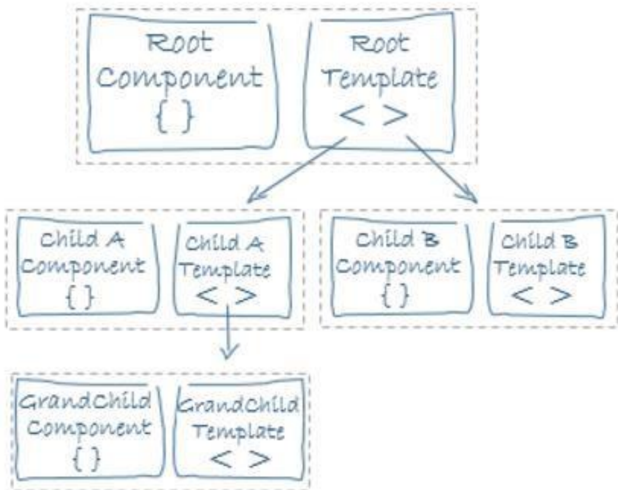
As described by the architecture diagram for an angular application there are 8 building blocks of Angular 2 application:

- Modules

An Angular module is simply a class decorated with @NgModule which is a function that takes one metadata object to describe the module by specifying properties like declarations(view classes), exports(declarations that can be made visible to other components), imports, providers etc.

- Components

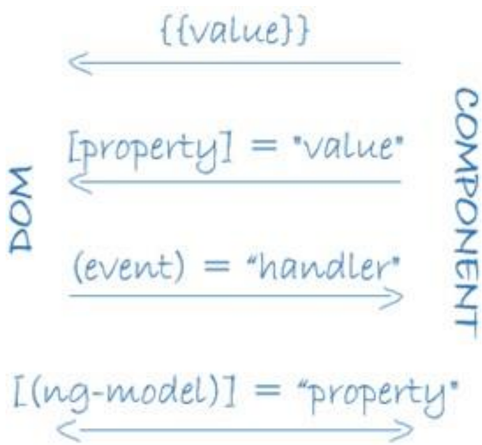
  Component is responsible for controlling a section of a screen known as view. Ever application has a component tree.



- Templates
- Metadata

  Metadata tells the application how a class is supposed to be processed. The metadata in @Components is where the building blocks of the component are specified. The template, metadata and component together come up to describe a view.
- Data Binding

  Refers to a mechanism for handling the coordination between different sections of a template within a component. The mechanism is can be clearly described from the picture below.



- Directives
- Services
- Dependency Injection

Primary Features of Angular.io:

- Angular is cross platform: This implies that it can be used to build high performance web apps as well as native mobile apps in combination with Ionic Framework, React Native etc.
- Angular gives enhanced performance and speed: Angular turns templates to a well optimized code for JavaScript virtual machines which enhances speed. It allows next to instant rendering of the application by using node.js or .NET or PHP to serve the view of the application. It supports 'code splitting' *i.e.* users only load code that is required and it is loaded with the Component Router of Angular.
- Modularity: A lot of core functionality has been to modules which helps in giving better speed.

DRF

Django REST Framework where REST stands for Representational state transfer is a powerful and flexible toolkit for building web REST API's for applications.

Django has been referred to as MVT (Model View Template) but Django follows the MVC Patterns closely enough to be called MVC framework.

When it comes to Django, Django views is considered to be controller and Django Templates to be views. In Django interpretation of MVC, the view describes the data that gets presented to the user.

Before understanding the backend flow in Django, let's understand **what ORM is?**

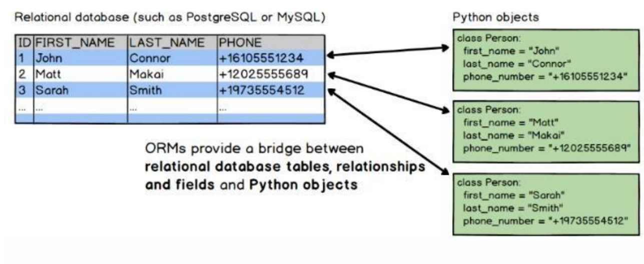
ORM Stands for Object Relational Mappers.

- The **Object** is the one developer use with python programming language.
- The **Relational** is the Relational Database Manager System. (A database, in the case of CandEx it is Maria DB).
- And finally the **Mapping** is where bridging is done with the objects and the database tables.

In applications where ORM framework is not used, bridging is done with the hand written code.

Briefly ORM is a technique that lets you query and manipulate data from a database using an Object Oriented Paradigm. An ORM Library can be written in any language of choice that encapsulates the code needed to manipulate the data. So user can directly interact with database using an object instead of writing long SQL commands.

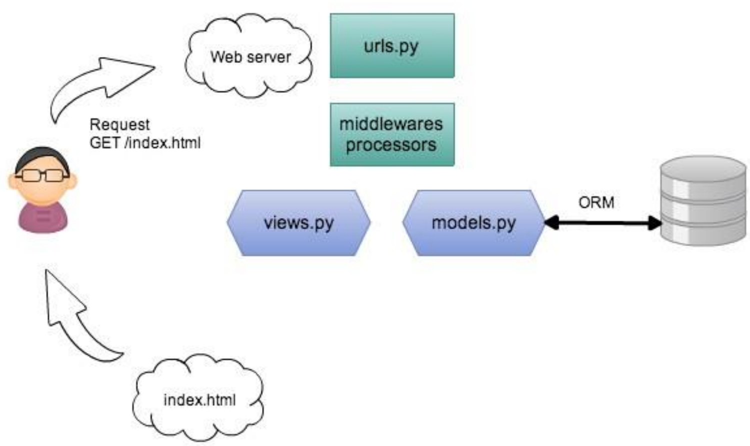
```
# This is an Object OF ORM
class Employee:
    def __init__(self,name):
        self.__name=name
    def getName(self):
        return self.__name
# This is a Table of Relational Database
create table employee(
        name varchar(10),
    )
# Using an ORM Framework Would allow developer to
# map that object with a relational database record
# something like this :
# Inserting employee into database
emp=Employee("Vincent")
orm.save(emp)
# Acessing employee using simple queries
emp1=Employee.query(name="Charles")
```



Workflow of Django:

All the actions performed by any user are CRUD operations (Create, Retrieve, Update and Delete). Each operation makes a HTTP request as explained in the below table:

OPERATION TYPE	REQUEST TYPE
Create	HTTP.post
Retrieve	HTTP.get
Update	HTTP.put
Delete	HTTP.delete



When a user requests for a page, application makes a HTTP request to Django server using an API URL.

On the Django side, urls.py contains the URL Routers for the REST API's. So when a request is made, a function written in views.py gets triggered which holds the complete business logic for an API. And views.py uses the concept of ORM (explained above) in order to access, update, create or delete the record in the database.

Git

Git is a version control tool or VCS (version control system) used to track changes in files and help in collaboration on same code between multiple people.

Primary Features:

- Support For non-linear development: it provides support for branching and merging at a high frequency and also facilitates in visualization of the same to provide a better idea of the non-linear development and its history. Branches are very lightweight and each one is a reference to just one commit.
- Support for distributed development: Git provides each user/developer its own local copy of the entire history of development and changes can be copied across repositories.
- Highly compatible: Repositories can be published via HTTP, FTP or over a plain socket or Secure Shell (SSH).
- Scalable to large projects: Git is faster than many other VCS in handling large projects and thus is considered to be highly scalable and fast.

Gulp.js

Gulp.js is JavaScript toolkit used in frontend web development as a task runner which is built on Node.js.It is built on node streams. It can compile sass files, compress js etc. The node streams allow file operation connections to be set up through ‘pipelines’.

Materialize

Materialize is UI component library that is built on CSS, JavaScript and HTML. It follows Material Design by Google.

Materialize is minimal and its color scheme is uniform across devices and platforms which makes it extensible and fit for personalization according to your own need. Materialize classes are created to fit screen size or in other words one can say that Materialize has in-built responsive design.

PrimeNG

PrimeNG provides rich UI components for Angular 2, quite similar to PrimeFaces which is a JavaServer Component Suite. PrimeNG is developed by PrimeTek Informatics. It is open source under MIT License.

PrimeNG provides about 70+ components to work with which are easy to use and accessible. PrimeNG incorporates responsive design with cross browser compatibility and is optimized for touch.

Node.js

Node.js is built on Chrome V8 JavaScript engine.

MariaDB

Maria DB is an open source relational database management system developed by the developer of MySQL itself. Maria DB was intended to maintain high similarity and compatibility with

MySQL. It used by giants like Google, Wikipedia and WordPress amongst others. It ensures ‘drop-in’ replacement for MySQL.

MariaDB uses more storage engines as compared to MySQL which include the following:

- Aria
- FederatedX
- OQGRAPH (MariaDB 5.2, MariaDB 5.3 and MariaDB 10.0)
- SphinxSE
- IBMDB2I. (MariaDB 5.5)
- XtraDB
- TokuDB (MariaDB 5.5, MariaDB 10.0)
- Cassandra (MariaDB 10.0)
- SEQUENCE (MariaDB 10.0)
- CONNECT (MariaDB 10.0)
- Spider (MariaDB 10.0)
- ColumnStore

6

System Design

Use Cases

Use case: Login

Actors:

Recruiter, HR team member

Triggers:

The user wishes to log in to CandEx.

Preconditions:

None

Post conditions:

- If the user enters correct credentials then he/she is logged in to the app and provided with an authentication token and redirected to the homepage of the application.
- If the user enters incorrect credentials then a notification appears intimating the user of failure to log in to the application and the state of the system remains the same.

Normal Flow:

- System will display the login screen which will require the user to enter username and password.
- User will enter username and password.
- The entered credentials are checked against the Users table in the database to check if the user is valid or not.

- If the entered credentials are valid then the user will be logged in and provided with an authentication token.
- Once logged in the user will be displayed the home screen, in this case the dashboard.

Alternate Flow:

- The user enters incorrect credentials so the verification fails and no authentication token is given to the user.
- The user is not redirected to other page and a simple notification informs the user that the credentials entered by the user are not valid.

Use Case: Raise a request

Actors:

Recruiter

Triggers:

The user wishes to create a new request.

Preconditions:

- The user must be a valid user and have the necessary privileges.
- The user must be logged in to the application

Post conditions:

- A new request will be created and added to the database.
- This new request will now be available for viewing and editing.

Normal Flow:

- The user will enter all the necessary fields according to his/her requirement.
- On clicking ‘Raise a Request’ button a notification pops up on the screen which informs the user of the successful creation of the request.
- On successful creation of a request the user is redirected to the ‘Manage Requests’ page.

Alternate Flow:

- If the user leaves certain mandatory fields the user is not allowed to proceed with submitting the form and the user is notified which field was incorrectly filled.

Use Case: List Requests

Actors:

Recruiter

Triggers:

The user wishes to view a list of preexisting requests.

Preconditions:

- The user must be a valid user and have the necessary privileges.
- The user must be logged in to the application

Post conditions:

- User will be displayed a page with the list of requests and some information related to a request.

Normal Flow:

- The user will click on ‘Manage Requests’ on the Navigation Menu which redirects them to the list of existing requests.
- A user can view the details of the existing requests and can wish to choose from a list of actions provided under the ‘Actions’ column.

Alternate Flow:

- If there exists no requests to start with, the user will be notified that presently there are no requests in the system.

Use Case: Edit request

Actors:

Recruiter

Triggers:

The user wishes to edit an existing request.

Preconditions:

- The user must be a valid user and have the necessary privileges.
- The user must be logged in to the application.
- The request to be edited should exist in the first place.

- The user must be on the ‘Manage Requests’ page.

Post conditions:

- The changes to the request are recorded and updated in the database.

Normal Flow:

- The user clicks on the edit button under the column of ‘Actions’ on the ‘Manage Requests’ page corresponding to the entry they wish to change.
- The user is redirected to a form with all the relevant fields which have prefilled information based on the present values of the request.
- User can access an input field and make necessary edits.
- If the form values are valid even after editing the form will be submitted and changes would be made in the database.
- A success message will appear on the screen to notify the user of the same.

Alternate Flow:

- If the user leaves certain mandatory fields the user is not allowed to proceed with submitting the form and the user is notified which field was incorrectly filled.
- If there is some other error, an error message appears on the screen informing the user that the changes made by the user were not successfully recorded.

Use Case: View Candidates Mapped to a Request

Actors:

Recruiter, Hiring Manager

Triggers:

The user wishes to see a list of candidates that are mapped to a particular request.

Preconditions:

- The user must be a valid user and have the necessary privileges.
- The user must be logged in to the application.
- The request to be under consideration should exist in the first place.
- The user must be on the ‘Manage Requests’ page.

Post conditions:

- The user is redirected to a page which lists candidates mapped to a request.

Normal Flow:

- The user clicks on the ‘View Mapped Candidates’ icon under the column of ‘Actions’ on the ‘Manage Requests’ page corresponding to the entry they wish to change.
- The user is redirected to a page which lists candidates mapped to a request.
- User can view details of these candidates.
- User can view Candidate name, email, Mobile number etc.

Alternate Flow:

- If the request has no candidates currently mapped to it the use will see an empty list and will also be notified of the absence of any candidates mapped to the requisition of interest.

Use Case: Add a Candidate

Actors:

Recruiter

Triggers:

The user wishes to add a new candidate to the system.

Preconditions:

- The user must be a valid user and have the necessary privileges.
- The user must be logged in to the application

Post conditions:

- A new candidate will be created and added to the database.
- This new candidate will now be available for viewing and editing.
- One can map this candidate to a request and start the process.

Normal Flow:

- The user will enter all the necessary fields according to his/her requirement.
- These fields capture a wide variety of data from candidate’s personal information like contact details and location to educational and professional information.

- The user can also enter the source and process start date of the candidate.
- The user can also upload a resume of the candidate using the application's file upload functionality.
- On clicking 'Submit' button a notification pops up on the screen which informs the user of the successful creation of the candidate.
- On successful creation of a candidate the user is redirected to the list of candidate's page which will allow the user to see the newly added candidate in the list.

Alternate Flow:

- If the user leaves certain mandatory fields the user is not allowed to proceed with submitting the form and the user is notified which field was incorrectly filled.

Use Case: List Candidates

Actors:

Recruiter

Triggers:

The user wishes to view a list of preexisting candidates.

Preconditions:

- The user must be a valid user and have the necessary privileges.
- The user must be logged in to the application

Post conditions:

- User will be displayed a page with the list of candidates and some information related to an each candidate.

Normal Flow:

- The user will click on 'Manage Candidates' on the Navigation Menu which will redirect them to the list of existing candidates.
- A user can view the details of the existing candidates and can wish to choose from a list of actions provided under the 'Actions' column which include 'Start Process', 'Ask for feedback', 'Schedule Interview', 'Offer details' etc.

Alternate Flow:

- If there exists no candidates in the system, the user will be notified that presently there are no candidates in the system.

Use Case: Edit Candidate

Actors:

Recruiter

Triggers:

The user wishes to edit information of an existing candidate.

Preconditions:

- The user must be a valid user and have the necessary privileges.
- The user must be logged in to the application.
- The candidate to be edited should exist in the first place.
- The user must be on the 'Manage Candidates' page.

Post conditions:

- The changes to the candidate are recorded and updated in the database.

Normal Flow:

- The user clicks on the edit button under the column of 'Actions' on the 'Manage Candidates' page corresponding to the entry they wish to change.
- The user is redirected to a form with all the relevant fields which have prefilled information based on the present values of the request.
- The user can switch tabs and choose from which information of the candidate they wish to change personal, educational, professional or other process related information
- User can access an input field and make necessary edits.
- If the form values are valid after editing the form will be submitted and changes would be made in the database.
- A success message will appear on the screen to notify the user of the same.

Alternate Flow:

- If the user leaves certain mandatory fields the user is not allowed to proceed with submitting the form and the user is notified which field was incorrectly filled.
- If there is some other error, an error message appears on the screen informing the user that the changes made by the user were not successfully recorded.

Use case: Send Mail to Candidate

Actors:

Recruiter

Triggers:

The user wishes to send a mail to the candidate in order to convey some information.

Preconditions:

- The user must be a valid user and have the necessary privileges.
- The user must be logged in to the application.
- The candidate to be to be notified should exist in the database.

Post conditions:

- The email is sent to the selected candidate using the SMTP server in the backend.

Normal Flow:

- The user clicks on the ‘Compose Mail’ icon under the column of ‘Actions’ on the ‘Manage Candidates’ page corresponding to the candidate they wish to send a mail to.
- The user is displayed a dialog box which contains a form with recipient field prefilled with candidate’s email and requires the user to fill other fields like subject and email body.
- The user can fill information according to their requirements.
- If the values filled by the user are valid then a mail is sent to the candidate and the user is notified with a success message.

Alternate Flow:

- If the user leaves certain mandatory fields the user is not allowed to proceed with submitting the form and the user is notified which field was incorrectly filled.
- If there is some other error, an error message appears on the screen informing the user that the changes made by the user were not successfully recorded.

Use case: Ask for Candidate Feedback

Actors:

Recruiter

Triggers:

The user wishes to ask for feedback regarding the interview process at the company.

Preconditions:

- The user must be a valid user and have the necessary privileges.
- The user must be logged in to the application.
- The candidate to be to be contacted should exist in the database.

Post conditions:

- An email is sent to the candidate which contains the URL that can be used to access the feedback form.

Normal Flow:

- The user clicks on the 'Ask for feedback' icon under the column of 'Actions' on the 'Manage Candidates' page corresponding to the candidate they wish to ask for feedback.
- On click a secret key is generated for the candidate and the URL is constructed with the candidate ID and his/her unique secret key.
- Further a mail is sent to the candidate which contains the above mentioned URL.

Alternate Flow:

- If the candidate has already been sent a mail with the URL to submit feedback, the user will be notified that the candidate has already been queried.
- If the candidate has already been sent the feedback and the candidate has submitted the feedback form then the user will be able to see the candidate's submission on clicking the same icon.

Use Case: View Master Table

Actors:

Recruiter, HR Team Member

Triggers:

The user wishes to view the contents of a master table.

Preconditions:

- The user must be a valid user and have the necessary privileges.
- The user must be logged in to the application

Post conditions:

- The user will be able to see the available master tables.

Normal Flow:

- The user will click on 'Masters' on the navigation menu
- The user will be displayed a page with initially only a select button where the user is required to select a master table that he/she wishes to view.
- On selecting an option from the list of existing master tables the user will be able to see a data table with the entries in the selected table.
- The user can change the select option and the data table will change accordingly.

Use Case: Edit a record in Master

Actors:

Recruiter

Triggers:

The user edit an existing entry in a master table.

Preconditions:

- The user must be a valid user and have the necessary privileges.
- The user must be logged in to the application
- The user has selected a particular master table from the list of master table.

Post conditions:

- User will be able to see the updated changes made to a record on successful change.

Normal Flow:

- The user will be able to see the list of records existing in the master table of interest.
- Beside the record that needs to be edited the user will click the 'edit' icon and this in turn will display a form on the right side of the data table that the user is working on.

- This form will be prefilled with current values of the entry and the user can change the field as per their requirements.

Use Case: Add a record to the master

Actors:

Recruiter

Triggers:

The user add a new entry in a master table.

Preconditions:

- The user must be a valid user and have the necessary privileges.
- The user must be logged in to the application
- The user has selected a particular master table from the list of master table.

Post conditions:

- User will be able to see the new record successful addition of the record.

Normal Flow:

- The user will be able to see the list of records existing in the master table of interest.
- Below the data table the user will be able to see an 'Add' button which upon clicking will display a form on the right side of the data table that the user is working on.
- The user can fill the required fields as per their requirement and click on 'Submit' button to save their changes.
- On submission and on passing all form validations the new record will appear in the data table and added to the database.

Alternate Flow:

- If the user leaves certain mandatory fields the user is not allowed to proceed with submitting the form and the user is notified which field was incorrectly filled.
- If there is some other error, an error message appears on the screen informing the user that the changes made by the user were not successfully recorded.

Use case: Add a User

Actors:

Admin

Triggers:

The admin wishes to add a user to the application.

Preconditions:

- The user must be a valid user of the application.
- The user must have admin privileges.
- The user must be logged in to the application.

Post conditions:

- A new user will be created and added to the system and mail to the new user will be sent about the same.

Normal Flow:

- The user will click on ‘Add User’ in the navigation menu to add a new user.
- On click, the user will be redirected to a webpage with a form which requires user to fill the username, password, email address first and last name of the new user they wish to add. The current user can give the, to be added user privileges and add them to certain groups.
- Upon filling the necessary fields, the user will click on the ‘Submit’ button to complete the process of adding a new user.
- If all form validations are passed then the user will be added to the system and will be able to log in the next time to the application, provided he/she has active status.
- An email will be sent to the newly added user intimating him/her their username and randomly generated password.

Alternate Flow:

- If the user leaves certain mandatory fields the user is not allowed to proceed with submitting the form and the user is notified which field was incorrectly filled.
- If there is some other error, an error message appears on the screen informing the user that the changes made by the user were not successfully recorded.

Use case: Edit a User

Actors:

Admin

Triggers:

The admin wishes to add a user to the application.

Preconditions:

- The user must be a valid user of the application.
- The user must have admin privileges.
- The user must be logged in to the application.

Post conditions:

- A new user will be created and added to the system and mail to the new user will be sent about the same.

Normal Flow:

- The user will click on ‘Add User’ in the navigation menu to add a new user.
- On click, the user will be redirected to a webpage with a form which requires user to fill the username, password, email address first and last name of the new user they wish to add. The current user can give the, to be added user privileges and add them to certain groups.
- Upon filling the necessary fields, the user will click on the ‘Submit’ button to complete the process of adding a new user.
- If all form validations are passed then the user will be added to the system and will be able to log in the next time to the application, provided he/she has active status.
- An email will be sent to the newly added user intimating him/her their username and randomly generated password.

Alternate Flow:

- If the user leaves certain mandatory fields the user is not allowed to proceed with submitting the form and the user is notified which field was incorrectly filled.
- If there is some other error, an error message appears on the screen informing the user that the changes made by the user were not successfully recorded.

Use case: List Users

Actors:

Admin

Triggers:

The admin wishes to view the existing users that can log in to CandEx.

Preconditions:

- The user must be a valid user of the application.

- The user must have admin privileges.
- The user must be logged in to the application.

Post conditions:

- A list of users with their groups and status as a user will be shown.

Normal Flow:

- The user will click on ‘List Users’ in the navigation menu to add a new user.
- On click, the user will be redirected to a webpage with a list of all the existing users the application with ticks and crosses to indicate their active, staff or superuser status and groups column.
- Upon clicking the information icon under groups a dialog box will appear which will display the groups the user is a part of, if any.
- Further in the dialog box a user can click on ‘View Permissions’ to see the permissions that that particular group has.
- If there is some other error, an error message appears on the screen informing the user that the changes made by the user were not successfully recorded.

Use case: Edit Profile

Actors:

Recruiter, Leadership, HR Team Member

Triggers:

The user wishes to change information related to his/her profile

Preconditions:

- The user must be a valid user of the application.
- The user must be logged in to the application.

Post conditions:

- The edited information of the user will be stored in the database.

Normal Flow:

- The user will click on ‘Edit Profile’ on the toolbar menu.
- On click the user will be displayed a dialog box containing a form with prefilled information of user’s first and last name and email address.
- A user can change his/her first name, last name or email address, however username once created cannot be changed for a user.

- On submission the changes will be updated in the database.

Alternate Flow:

- If the user leaves certain mandatory fields the user is not allowed to proceed with submitting the form and the user is notified which field was incorrectly filled.
- If there is some other error, an error message appears on the screen informing the user that the changes made by the user were not successfully recorded.

Use Case: Change Password

Actors:

Recruiter, Leadership, HR Team Member

Triggers:

The user wishes to change their current password.

Preconditions:

- The user must be a valid user of the application.
- The user must be logged in to the application.

Post conditions:

- The password will be changed successfully.

Normal Flow:

- The user will click on ‘Change Password’ on the toolbar menu.
- On click the user will be displayed a dialog box containing a form which will require the user to fill their username and current password to verify their credentials and to fill their new password twice.
- On submission and on passing all form validations the user will be informed via a notification if the password change was successful.
- On submission the password will be saved.

Alternate Flow:

- If the user leaves certain mandatory fields the user is not allowed to proceed with submitting the form and the user is notified which field was incorrectly filled.

- If there is some other error, an error message appears on the screen informing the user that the changes made by the user were not successfully recorded.
- If the username and current password entered do not match the user will be notified and no password change will take place.
- If the new entered password entered do not match the user will be notified of the same and asked to try again.

Use Case: Create a new email template

Actors:

Recruiter

Triggers:

The user wishes to add a new email template.

Preconditions:

- The user must be a valid user and have the necessary privileges.
- The user must be logged in to the application

Post conditions:

- A new email template will be created and added to the database.
- This new email template will now be available for viewing and editing.

Normal Flow:

- The user will enter all the necessary fields according to his/her requirement.
- On clicking ‘Save’ button a notification pops up on the screen which informs the user of the successful creation of the email template.
- On successful creation of an email template the user is redirected to the ‘Manage Email templates’ page.

Alternate Flow:

- If the user leaves certain mandatory fields the user is not allowed to proceed with submitting the form and the user is notified which field was incorrectly filled.

Use Case: List Email Templates

Actors:

Recruiter

Triggers:

The user wishes to view a list of preexisting email templates.

Preconditions:

- The user must be a valid user and have the necessary privileges.
- The user must be logged in to the application

Post conditions:

- User will be displayed a page with the list of email templates and some information related to them.

Normal Flow:

- The user will click on 'Manage Email Templates 'on the Navigation Menu which redirects them to the list of existing email templates.
- A user can view the details of the existing email templates and can wish to choose from a list of actions provided under the 'Actions' column.

Alternate Flow:

- If there exists no requests to start with, the user will be notified that presently there are no requests in the system.

Use Case: Edit email template

Actors:

Recruiter

Triggers:

The user wishes to edit an existing email template.

Preconditions:

- The user must be a valid user and have the necessary privileges.
- The user must be logged in to the application.
- The request to be edited should exist in the first place.
- The user must be on the 'Manage Email Templates' page.

Post conditions:

- The changes to the email template are recorded and updated in the database.

Normal Flow:

- The user clicks on the edit button under the column of ‘Actions’ on the ‘Manage Email Templates’ page corresponding to the entry they wish to change.
- The user is redirected to a form with all the relevant fields which have prefilled information based on the present values of the request.
- User can access an input field and make necessary edits.
- If the form values are valid even after editing the form will be submitted and changes would be made in the database.
- A success message will appear on the screen to notify the user of the same.

Alternate Flow:

- If the user leaves certain mandatory fields the user is not allowed to proceed with submitting the form and the user is notified which field was incorrectly filled.
- If there is some other error, an error message appears on the screen informing the user that the changes made by the user were not successfully recorded.

Use case: Adding an Interview Level

Actors:

Recruiter

Triggers:

The user wishes to create a new interview level for a candidate.

Preconditions:

Interview Process for the candidate has been started.

Post conditions:

New Interview Level will be created for a particular candidate.

Normal Flow:

- On a click of schedule button corresponding to that candidate, user is welcomed with a form for creating a new level.
- Interview Level form checks for the existing levels created for that candidate and provides the option for the level number accordingly. Level Form block the dates less than the current date in the date picker for the interview level date.

- User can also set the level as the final level by selecting the check box.
- The entered details are stored in the database and a new level is created with level status as ‘Scheduled’.
- And finally user is asked to add Interview rounds for the new level created.
- If the interview Level date matches the current date, a form gets added asking user if the candidate has appeared or not. User is provided with three buttons:
    1. YES : This button sets the level status to ‘Started’
    2. NO: This button set the level status to ‘Not Showed’ and stores the current date as the First No-Show in the database.
    3. NOT YET: This button waits for the user to press ‘YES’ or ‘NO’ button for a final call.

Alternate Flow:

- If the user fails to enter the correct details for a level and adds the level mistakenly or want to reschedule the level for candidate’s second attempt, he/she can update the newly created level by clicking on an edit button.
- After a Second No-Show of the candidate, user is asked to drop the candidate.

Use case: Adding an Interview Round

Actors:

Recruiter

Triggers:

The users wishes to add a new interview round for an existing interview level.

Preconditions:

- An Interview level should be existing in the database for a candidate.
- List of panel members should be existing in the database.

Post conditions:

- A new interview round will be created for that particular level.
- And a mail is sent to the panelists added to the newly created round with all the interview round information and a feedback URL in it.

Normal Flow:

- On a click of ‘Add New Round’ User is presented with an interview round form.
- Round form blocks all the date less than the current date in date picker and validates that the interview start and finish time entered by the user are greater than the current time.
- User can add the multiple panelist for a round.
- After entering the required details for the new round and clicking the ‘Schedule’ button. The entered details are stored in the database and a new round is created for that level with the round status as ‘Scheduled’.
- Along with a new round creation, a mail is sent to all the panelists associated with that particular round informing them about the interview round information and a feedback URL

Alternate Flow:

- If the user fails to enter the correct details for a round and adds the round mistakenly or want to reschedule the round, he/she can update the newly created round by clicking on an edit button.

Use case: Feedback URL Generation For each Panelists

Actors:

Triggers:

Each of the panelist is sent a feedback URL, generated right after the round is scheduled.

Preconditions:

- List of panel members should be existing in the database.
- When a round is created or rescheduled.

Post conditions:

A feedback URL is generated for each of the panelists.

Normal Flow:

- When a panelist is added to a round a secret key is generated using combination of random numbers and alphabets.
- Each panelist is assigned a generated secret key.
- And the same secret key is appended to the feedback URL as a query parameter.

- This feedback URL is sent to panelist via mail.
- This URL results into a web page that has 4 Separate tabs:
    1. Interview Feedback & Notes
    2. Interview Process Timeline
    3. Other Panelist Notes
    4. Candidate Resume

Use case: Note Submission by panelists

Actors:

Panelists

Triggers:

When the user is taking an interview and writing down the notes to submit in the end of the interview.

Preconditions:

- A Mail is received by the panelist about the interview round scheduled and a feedback URL in it.

Post conditions:

- Interview finish time is updated and the round status is set to 'Completed'.
- Notes added by the panelist are stored in the database and are made available to the other panelists of all the levels for a candidate in their feedback URL.
- A mail is sent to the panelists of next round reminding about interview that has to be taken by them.
- User is presented with the feedback form after the notes form is submitted.

Normal Flow:

- When a user clicks on the Feedback URL sent to his email address, user is presented with a notes form including some basic information about the candidate to be interviewed.
- During the interview time, panelist writes down the notes (about what topic has been covered during the interview hour) using the editor present in the form.
- When the interview is done user sets the interview finish time by clicking on the button 'NOW'. And clicks the button 'I'm Done' in order to submit the notes form.

- Interview round status is changed to ‘Completed’.
- And an interview reminder is sent to all the panelists of the next round.
- And in the last the user is show the feedback form which can be submitted later as well.

Alternate Flow:

- If the user tries to access the feedback URL prior to the interview date and start time, he is informed that the link will only be active after the aforementioned interview data and time.
- If the user tries to change the feedback URL query parameters in order to have an access to other panelist’s feedback URL, he is redirected to 404 (Not Found) page being an invalid user to their feedback URL
- If the user fails to submit the notes form right after the interview completed, he/she can reopen the feedback URL and enter the interview finish time manually using the time picker and submit the form.

Use case: Feedback Submission by panelists

Actors:

Panelists

Triggers:

After the notes submission, when user wishes to submit the interview feedback.

Preconditions:

- Interview is taken and
- Notes submission has been done successfully.

Post conditions:

- Feedback form containing the panelist’s decision and comments will be stored in the database.
- Resubmissions are not accepted.

Normal Flow:

- When a user decides to fill the interview feedback form right after the interview is taken or later, he is presented with the candidate basic information and a separate section for his comments and decision about the interview he took.

- After the feedback form submission, panelist feedback is made available to the recruiter in order to make the final decision for the round.

Alternate Flow:

- If the user tries to submit the feedback form for the second time by accessing the Feedback URL again, he is welcomed with previous displayed feedback.

Use case: Displaying Candidate Resume to Panelists

Actors:

Panelists

Triggers:

When the users tries to look for candidate resume on feedback URL

Preconditions:

Candidate Resume was uploaded by the recruiter while adding the candidate.

Post conditions:

- If the resume uploaded has a file format of .pdf, it will be viewed using pdf viewer.
- And if the file format is .doc or .docx a link is provided to download the file.

Normal Flow:

- When a user tries to access the feedback URL before or after the interview date and time, Resume tab will always remain active.
- On selecting the resume tab, user can view or download the resume depending on the file format.

Use case: Setting Final Decision for a Round

Actors:

Recruiter

Triggers:

When the users wishes to submit the final decision for an interview round.

Preconditions:

- Interview round has been held and round status is ‘Completed’.

- All the panelists have submitted their own decision and comments in the feedback form.

Post conditions:

- Round decision will be changed to ‘Recommend’ or ‘Not Recommended’.
- If the submitted decision for the round is the final round of a level, user is asked to enter the final decision for the level with that round.

Normal Flow:

- When a user accesses the ‘Schedule Interview’ Page for a particular candidate and checks the interview rounds for a particular level.
- He/she can view the feedback from the panelists for all the rounds held and check for their decisions and comments about the candidate.
- And can set the final decision for the rounds as ‘Recommend’ or ‘Not Recommended’.
- If all the round’s decision has been filled for a level, user is displayed with an additional form for filling the level decision.

Use case: Setting Final Decision for a Level

Actors:

Recruiter

Triggers:

When the users wishes to submit the final decision for an interview level.

Preconditions:

- All the interview rounds are held with the status as ‘Recommend’ or ‘Not Recommended’.

Post conditions:

- Level decision will be changed to ‘Cleared’ or ‘Not Cleared’.
- If the level cleared is a final level, user is asked to set the hiring decision date.

Normal Flow:

- User enters the ‘Schedule Interview’ page for a particular candidate and finds all the rounds status to be ‘Recommend’ or ‘Not Recommended’.
- User is displayed with a small form for setting the level decision with options as ‘Cleared’ and ‘Not Cleared’.

- When user fills the required fields and submits the form, level decision is updated in the database.
- If the level is a final level for a candidate, user is asked to set the hiring decision date.

Alternate Flow:

- If user tries to set the level decision to ‘Cleared’ or ‘Not Cleared’ before all the rounds status are updated to ‘Recommend’ or ‘Not Recommended’.
- A pop up message appears alerting the user to enter all the round decisions before setting the final decision for the level.

Use case: Setting the Hiring Decision Date

Actors:

Recruiter

Triggers:

When the user wants to set the hiring decision date of an interview process for a particular candidate.

Preconditions:

- All the interview levels have a level decision of ‘Cleared’ especially the final level.

Post conditions:

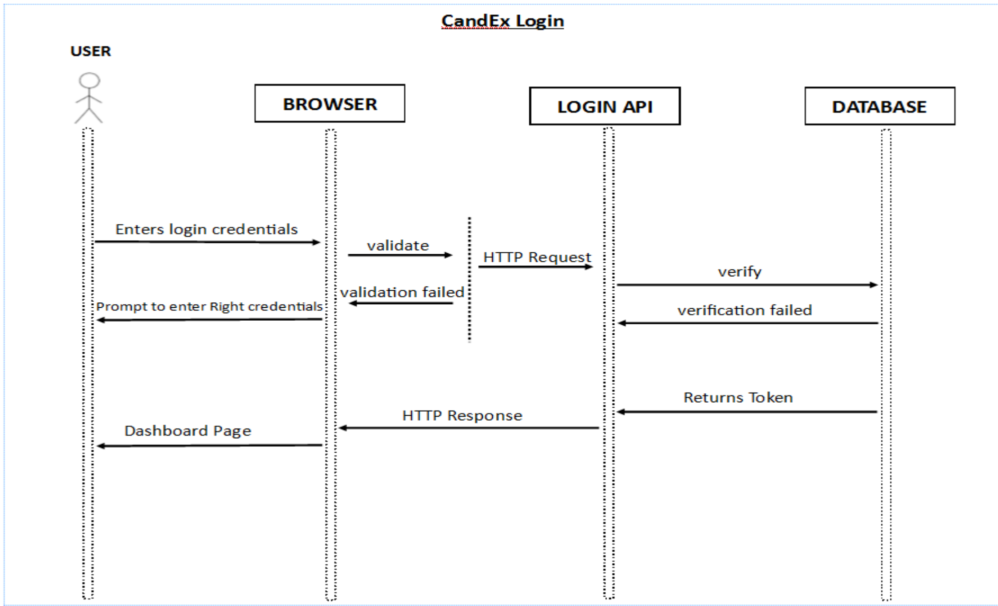
- Hiring Decision date is set for the candidate.
- And the user can now enter the Offer details.

Normal Flow:

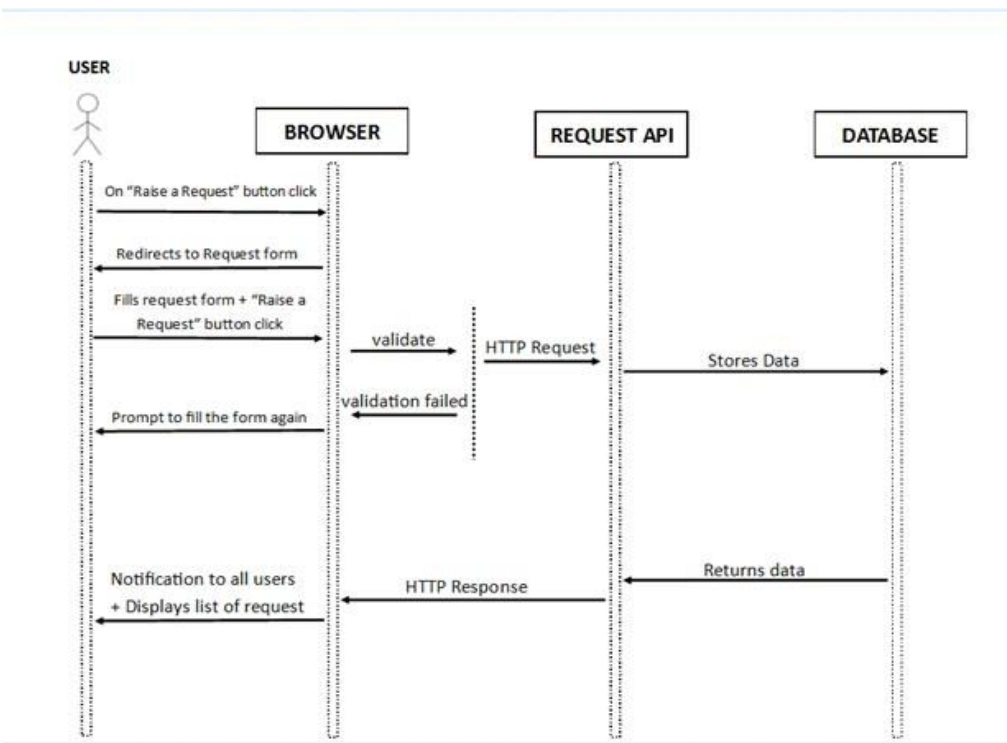
- User enters the ‘Schedule Interview’ page for a particular candidate and finds a form prompting him/her to set the hiring decision date.
- Form block the dates less than the current date in the date picker.
- After the user has set the hiring decision date, it is stored in the database.
- And the user is provided the access to fill offer details for the candidate.

Sequence Diagrams

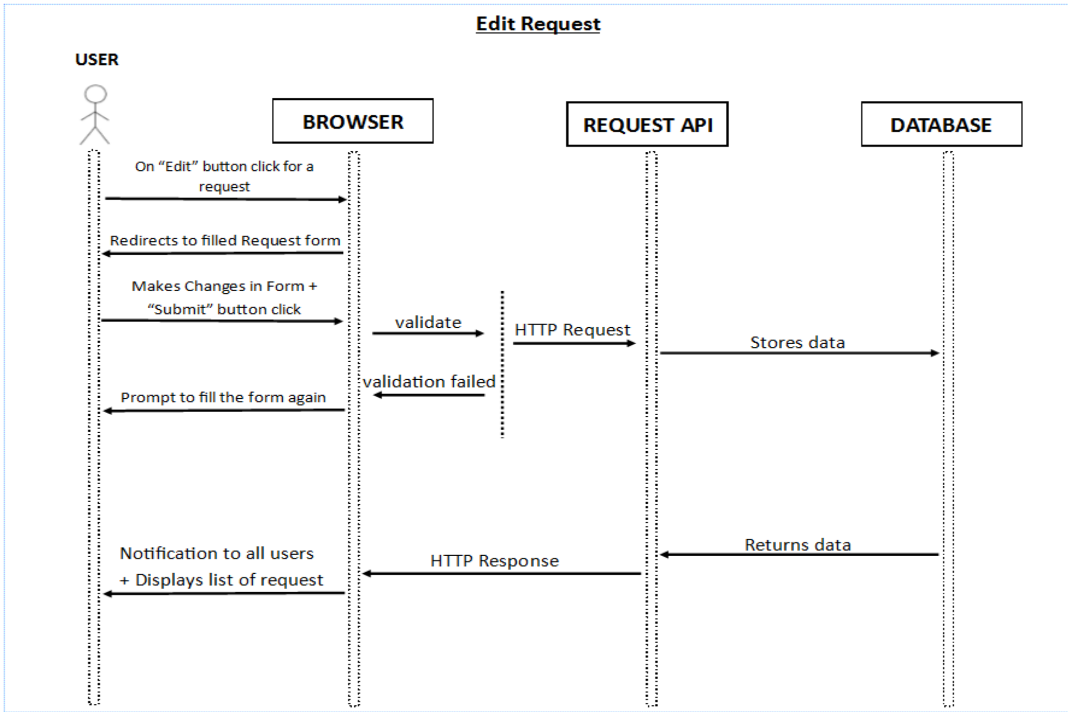
Login



Request Creation



Editing a Request


Edit Request
USER
BROWSER
REQUEST API
DATABASE
On "Edit" button click for a request
Redirects to filled Request form
Makes Changes in Form + "Submit" button click
validate
HTTP Request
Stores data
validation failed
Prompt to fill the form again
Returns data
Notification to all users + Displays list of request
HTTP Response

7

Features

Profile Management and Authentication

CandEx users can log in by providing the correct set of credentials for their username and password. CandEx maintains a table of users in the database with the list of users with their usernames, first and last name, email address and hashed password. If someone attempts to log in, this user table is checked if such a user exists and if they have provided the correct set of credentials. If the user is a valid one, the user is logged in to the application and redirected to the homepage.

There exists a superuser or admin account which has the power to create more users. The admin can simply go to the option of 'Add a User' and provide a username, randomly generated password and other information to add a user. Upon confirming the creation an email will be sent to the newly created user with information that'll allow them to log in to the application. On successful creation the user is added to the users' table and can access the application.

A user can choose to edit their profile where they have the flexibility to change their first name, last name and/or their email address and the changes made will be updated in the aforementioned users table.

The user can also change their password. This process requires the user to first confirm their current username and password, and if that is valid then the process of changing the password is carried forward. The user is required to fill their new password, twice to avoid errors. If the two places where the new password is typed matches, the password of the user is updated.

Apart from users of the application, CandEx also exposes certain pages to one-time users that have highly restricted access to a page or a set of pages that can be reached through via a link, such users cannot access other pages and will not be able to see what other users of CandEx can see due to security reason as we don't want to give all privileges to somebody who doesn't require it and may pose potential threat to the integrity of the existing database. For such a case a

URL is generated on fly and sent to the intended one-time user via an email, they can access it and view/edit the page or set of pages available to them. They might be able to revisit these pages but not any other page of the application that they are not authenticated to.

CandEx authentication system uses token based authentication. So when a successfully logs in to the application there are given a token which is saved in a cookie which allows them to access different parts of the application. If the user logs out, this cookie is destroyed and they can no longer get inside the application. Token authentication is considered fit for client-server setup of applications like CandEx.

Requisition and Candidate Management

CandEx provides support for requisition and candidate management. This involves creation of a request and the flexibility to edit it as and when required.

Requisition Management

'Raising of a request' involves providing certain mandatory details and on passing all the necessary validations a request is created and added to the system. A user can view the existing requests in the database and their corresponding details like status of request, whether it is open or closed, and the department and designation associated with the request. There are also additional fields like experience and position applied that can be seen. One can also see the candidates mapped to a particular request by choosing the appropriate option provided along with the list of existing requests.

Candidate Management

Our application captures a wide variety of candidate information for a candidate and provides a diverse set of action for a candidate record in the system as the idea of the application is meant to be centered around a candidate, thus the name 'Cand'-Ex.
For a candidate the following information is captured:

Personal: Candidate's first name, last name, gender, his/her contact details which includes a phone number and email address, their present and native cities.

Educational: Multiple rows can be added for different level of education i.e. graduate, post graduate or doctorate level. For each row the user is supposed to provide the institution, course name grade point average and graduation type.

Professional: For a candidate, details like current employer, designation and CTC and total work experience are captured.

Process Related Details: These details are details that are specific to a candidate's interview process at ViaSat India. Under this fields like process start date, team, title etc. are recorded.

Resume: CandEx allows the user to also upload the resume of an employee either in pdf or doc format which can be used for viewing later on.

One can add a candidate to the system by providing the mandatory information out of the fields mentioned above. Once added the status of a candidate is set as active and a candidate must be mapped to a particular request at the time of creation itself.

Similar to requisitions, the user can view the list of candidates in the system and their details. The user can further edit the details of these preexisting candidates.

User can schedule interviews for a candidate by adding different interview levels and various rounds under each level as per the interview requirement for the process of hiring. This is a step by step process. At first a level has to be added for a candidate where the user needs to select the level name and the date of level and a checkbox needs to be selected if the level is to be considered as the final level.
Upon creation of a level the user/recruiter can add a round which will essentially be equivalent to scheduling an interview for the candidate. Under this, the user needs to provide the list of panelists that will be taking the interview, mode of interview, date, start time and estimated end time for an interview. Upon confirmation the round details will be recorded and added to the candidate's profile. Similarly, at a time multiple rounds and levels can be added for a candidate. The application also records the status of candidate no-shows so in case a candidate does not show up for an interview it can recorded in the schedule interview page itself by selecting a radio button. If the candidate fails to show up for the very first interview itself, the candidate is automatically dropped.

Further for a candidate that has completed his/her interview process and has received positive feedback, his/her hiring decision date upon finalization can be captured by the application. On setting the hiring decision date for a candidate the following details are required to be filled:

Offer Details-This involves fields joining and offer date for a candidate. Additionally, there is an option of updating the offer status whether the offer was accepted, declined or put on hold by the candidate.

Cost Details- Once the candidate has been extended an offer and the cost details table is available to make changes in for that candidate as if the candidate does join the user will require the cost the company spent on hiring that particular person. This will include different kinds of costs like relocation, settling-in, agency cost and the salary of the candidate.

Upon the confirmation of acceptance of offer by a candidate the offer status can be updated to 'Accepted' for that candidate. This will enable two more features- Candidate Engagement and Employee Details.

Candidate Engagement Plan-According to the notice period of a candidate the candidate engagement is laid down by the recruiter. Candidate engagement is simply a plan with a set of actions or tasks that need to be carried out during the notice period of a candidate to intimate the candidate about the culture at the company and make the transition from a candidate to an employee smooth. So the user can add tasks and select the due dates for each task and lay out a plan for each candidate accordingly. Am email will be sent to person responsible for carrying out these tasks as a reminder 2 days before the due date which will be an automated process.

Now once the offer has been accepted there are certain details that need added for a candidate that are specific to their status as a ViaSat employee like employee ID, user id, manage ID designation etc. These are captured under a section called 'Employee Details' that is made available only on the confirmation of the onboarding of the candidate.

The user can also wish to ask candidate for a feedback for the interview process which will send an email to the candidate where he/she can fill the feedback form which is a one-time submission only and the candidate’s feedback will be recorded in the system and can be viewed later on using the same icon for candidate feedback.

Email Functionality

CandEx also takes care of sending emails internally within ViaSat and to candidates as well. The list of emails, their intended recipients and their respective triggers are listed below:

Email Type	Recipient	Event Trigger
Request Creation	Recruiter, Hiring Manager	On successful creation of a request.
Request Updation	Recruiter, Hiring Manager	On successfully editing a request
Interview Scheduled	Interview Panelists	On scheduling of an interview
Candidate Feedback	Candidate	On clicking the option for ‘Ask for Feedback’ for a particular candidate.
Candidate Engagement Reminder	Responsibility Holder	Sent automatically 2 days before the due date of a task.
Interview Round Feedback Reminder	Interview Panelists	Reminder mail sent to the interview panelists if they have not submitted a feedback for a candidate.

Reports and charts

Given the large amount of data that CandEx captures it can also provide them a look at certain metrics that will give the HR department and the leadership a deeper insight into the hiring process of the company and take steps to make it a better experience for the candidate and as well to make requisition closing faster by finding candidates best fit for a position.

The reports and charts that CandEx allows its users to view are listed below:

1. Candidate process time at each level – Gives an average of time taken for candidates at different levels.

2. Requisition Ageing – Displays the age of an open requisitions in days in form of a bar chart.

3. Future Starts – Considers a span of four months and reports the number of people that are expected to join the company as employees for each month. This report is visualized in form of a bar chart.

4. Diversity Mix for Candidates – Based on the source and gender of a candidates the diversity mix is presented.

5. Active requisitions with candidates' pipeline- Reports the number of candidates mapped to presently open requisitions.

6. Weekly interview time slots- Displays the interview slots for each panel member and how many hours they dedicated to the task of taking interviews and for which levels.

7. Cost per hire- This report gives estimate of cost per hire on the basis of different categories for any candidate who has accepted the offer and is expected to join as an employee.

8. Education Reports – This report shows a classification of how candidates are distributed according to their graduation type *i.e.* graduate level, post graduate level and doctorate.

9. Past Employer- A report describing the past employers of candidates in the pipeline.

10. Hiring Manager Based Requisitions- Reports the number of active requisitions on the basis of hiring managers.

11. Recruiters Based Requisitions – Reports the number of active requisitions on the basis of recruiters that opened them.

12. Notice period to join – Reports the notice period for all the candidates that have been extended an offer.

8

Conclusion

CandEx removed the hassle of maintaining large excel sheets that were tough to manage when the data expanded and added redundant work for the recruiters by providing a platform where all the data could be captured and that will automatically generate reports and metrics for its users.

It reduced a lot of manual work by automating the task of sending email reminders that was earlier a task that the recruiter had to do.

CandEx also allowed the people from the leadership team to simply access reports they were interested in by using a particular URL and removed the process of sharing the same data via mail which involved the people from the HR department to compile various reports and then send it across every month.

References

- Brochere: ‘Looking for top talent? Move to mobile recruiting’, IBM Coroporation
- IBM Kenexa BrassRing on Cloud
- Git Official Documentation, https://git-scm.com
- Angular.io Officaila Documentation, https://angular.io/
- http://materializecss.com/
- www.django-rest-framework.org