KINGDOM OF SAUDI ARABIA
Ministry of Higher Education
Al-Imam Mohammad University
College of Computer & Information
Sciences

المملكة العربية السعودية
وزارة التعليم العالي
جامعة الإمام محمد بن سعود الإسلامية
كلية علوم الحاسب والمعلومات

Second term 1441/2020

Software Engineering (CS- 310)
BSCS- Section: 171

Project-Phase No: 2
Smart Job Interviewer
(Design document)

Submitted By

Mansour Abdulaziz Alsaleh (440029449)-Coordinator

Abdulrahman Saad Almutairi (439026126)

Abdullah Abdulrahman Albkhit (439013683)

Yasser Abdullah Alsamel (439012305)

Firas Abdullah Alhussain (438011674)

Supervisor

Sultan Saud Al Qahtani

Date: 11/11/2020

Table of Contents

1. Introduction

The “Smart Job Interviewer” is a website that companies use for writing the description of the applicant they need. By writing a set of questions the applicants need to answer. The website should be available on the company’s webpage whenever the company needs more applicants.

Employers should write a set of questions and give it a timeline. In that timeline the applicants can enter the website and answer the questions based on their experience and skill. After the timeline ends, the employer will have a good overview on the applicants he received. Then he/she will pick the best applicant/applicants for the job.

Applicants can also browse the website to find companies that are hiring. The SJI has a search system for applicants to use.

In conclusion, this software requires the users to have an internet connection in order to access the website and use it.

Progress: The progress made from the SRS document is a lot, but the key points are the functional requirements which we have finished in phase #1. It plays a huge part in this phase as it is the base of the design phase with all the functional requirements set and ready, the work on the design phase should be easy and simple.

2. Design

2.1 System-Level Diagram

The architectural pattern, as shown in Figure 1. It is composed of subsystems with each subsystem relating to each other by links. And the use case, as shown in Figure 2. Is to show how system user (actors) can interact with the system’s function and their relations with the other users’ functions.

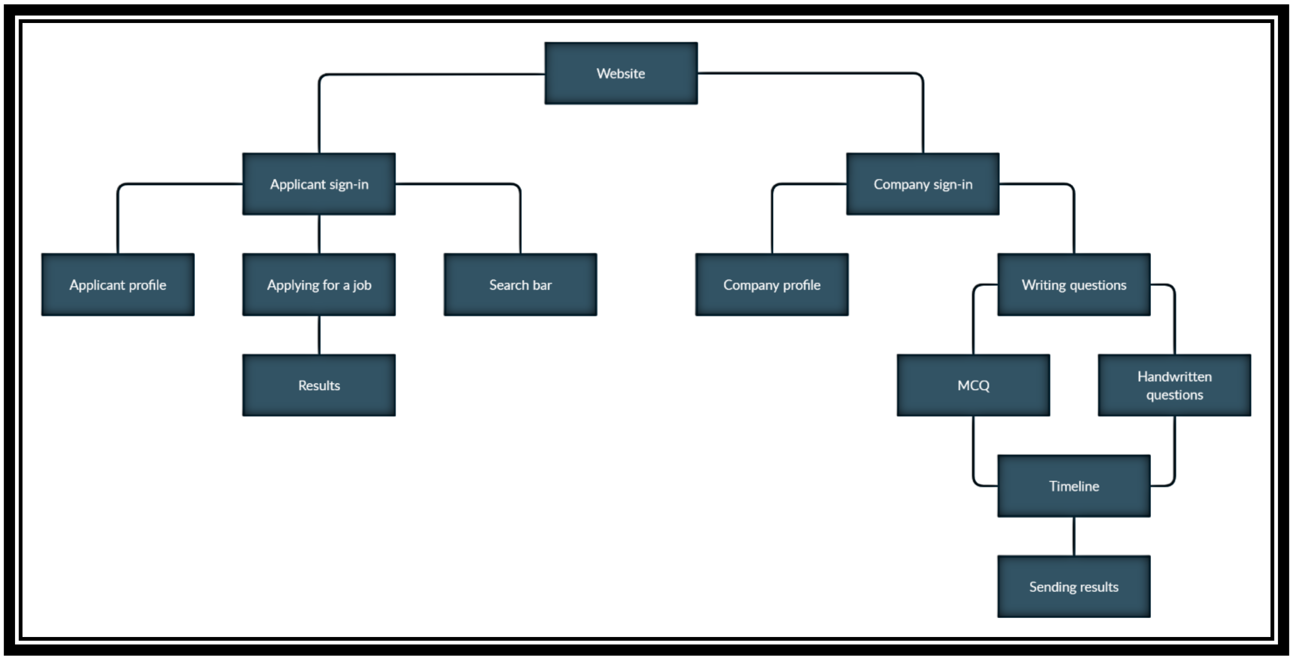

Figure 1 SJI System Architecture

Subsystems that need to be described:

Applying for job

Is the way for applicants to apply for a job but the company must be hiring. After the applicant finds a company that are hiring then he/she can apply and start answering questions.

Results

After the applicant has applied and finished answering the questions, the applicant will receive an email at the end of the timeline indicating if he/she got accepted to the company or not.

Writing questions

For the employer to hire applicants he/she must write the questions. The questions can be MCQ or handwritten**.**

MCQ

The MCQ’s are written by the employer and given a weight for calculating the result automatically. That helps wit time and effort.

Handwritten

The handwritten questions are written by the employer and this type of question must be given a weight manually. Therefore, the employer must go over every question of this type and give it a weight.

Timeline

The employer must set a timeline for the beginning and the end of the hiring period. Also, the employer must set a time for answering questions.

Sending results

After the hiring timeline ends, the employer would then pick the best candidates for the job and send them an email indicating that an interview will be done with them. If all goes well in the interview, then the employer would send them an email indicating they got accepted. And also send rejection emails to those who didn’t get accepted.

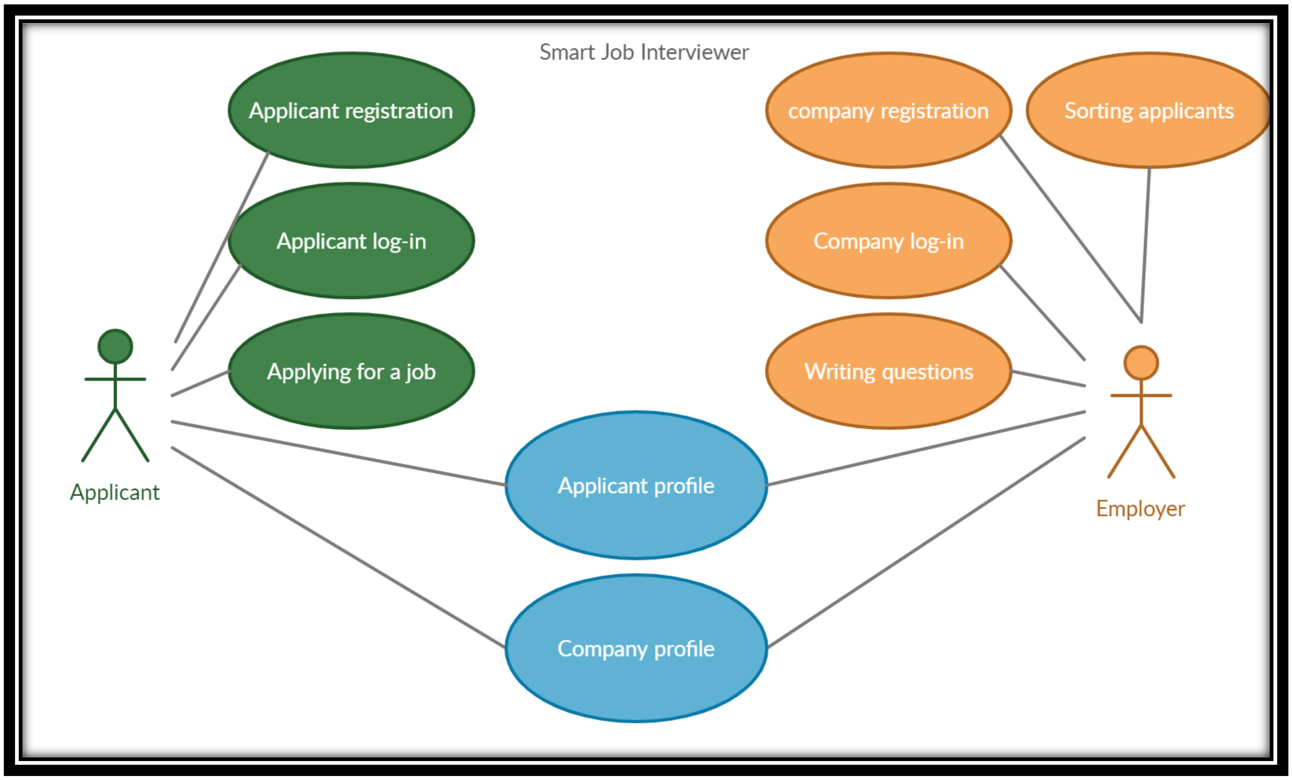

Figure 2 SJI use cases

2.2 Class Diagram

In the class diagram section, the classes and methods of the SJI will be shown and briefly described.

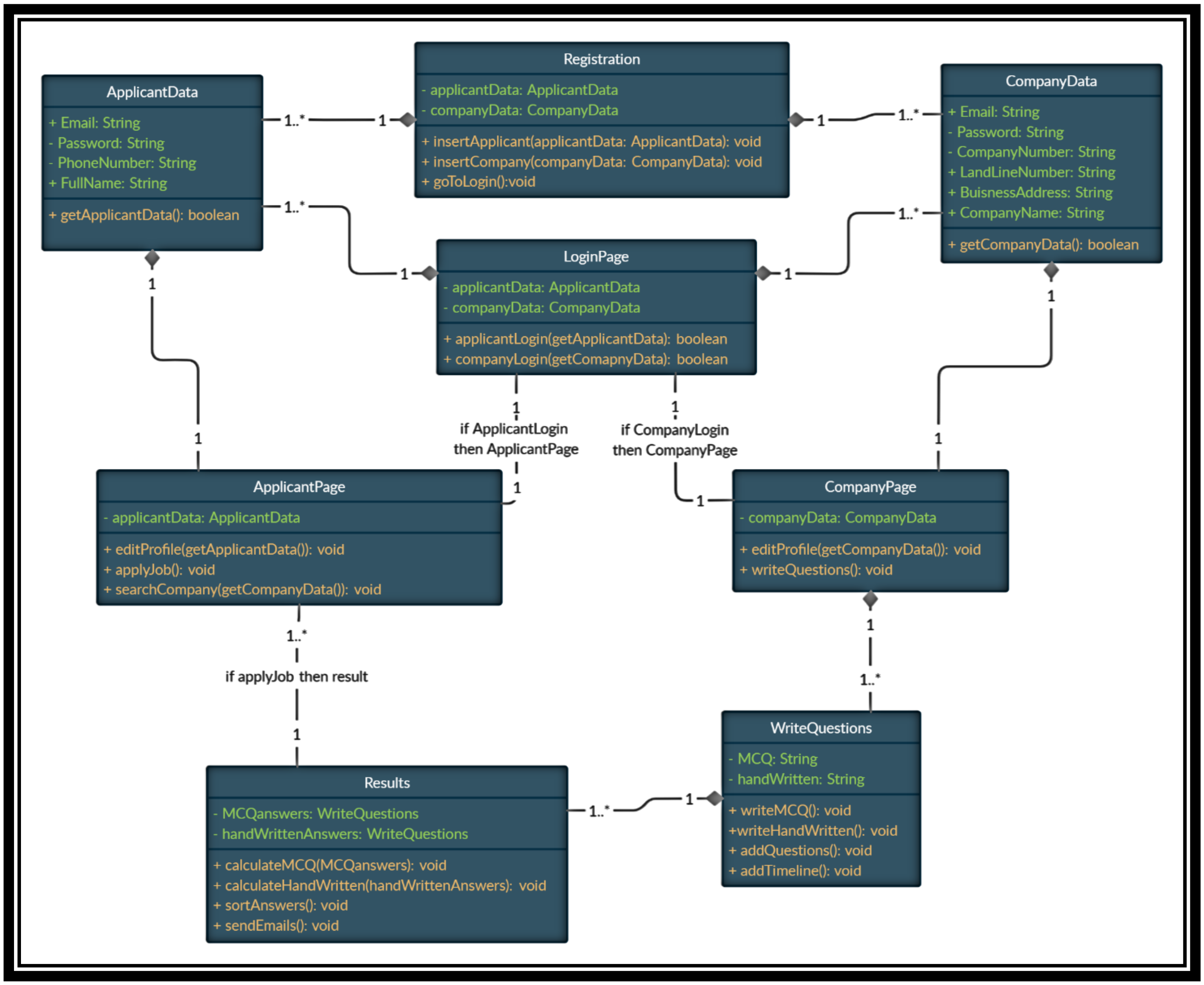
Registration
- applicantData: ApplicantData
- companyData: CompanyData
+ insertApplicant(applicantData: ApplicantData): void
+ insertCompany(companyData: CompanyData): void
+ goToLogin():void
ApplicantData
+ Email: String
- Password: String
- PhoneNumber: String
+ FullName: String
+ getApplicantData(): boolean
CompanyData
+ Email: String
- Password: String
- CompanyNumber: String
+ LandLineNumber: String
+ BuisnessAddress: String
+ CompanyName: String
+ getCompanyData(): boolean
LoginPage
- applicantData: ApplicantData
- companyData: CompanyData
+ applicantLogin(getApplicantData): boolean
+ companyLogin(getComapnyData): boolean
if ApplicantLogin then ApplicantPage
if CompanyLogin then CompanyPage
ApplicantPage
- applicantData: ApplicantData
+ editProfile(getApplicantData()): void
+ applyJob(): void
+ searchCompany(getCompanyData()): void
CompanyPage
- companyData: CompanyData
+ editProfile(getCompanyData()): void
+ writeQuestions(): void
if applyJob then result
WriteQuestions
- MCQ: String
- handWritten: String
+ writeMCQ(): void
+writeHandWritten(): void
+ addQuestions(): void
+ addTimeline(): void
Results
- MCQanswers: WriteQuestions
- handWrittenAnswers: WriteQuestions
+ calculateMCQ(MCQanswers): void
+ calculateHandWritten(handWrittenAnswers): void
+ sortAnswers(): void
+ sendEmails(): void
1
1..*

Figure 3 SJI Class Diagram

2.3 Class Method Description

The classes and methods will be described in this section.

1. Results Class

In this class, the results of the applicant’s questions are calculated and sorted. The MCQ questions are calculated automatically by adding the point of every question and giving the sum. Then the employer would calculate the handwritten questions manually and give the points which will add up with the MCQ points and give the sum. Finally, the sum of every applicant will be sorted from best to worst.

Results Methods

Method 1: calculateMCQ

Description: This method adds the total amount of points that the applicant has answered by taking the applicants answers and comparing them to the employer’s model answers and gives the sum at the end.

Method 2: calculateHandWritten
Description: This method lets the employer give the questions points and then the method automatically adds up all the points (with the MCQ points) to give the sum at the end.

Method 3: sortAnswers
Description: This method sorts all the answers from best to worst based on the sum of the answers.

Method 4: sendEmails
Description: After the timeline ends, the employer will choose the best candidates then this method will send interview dates to the selected candidates. If the interviews have gone well and the employer does not need more applicants, then this method will send rejection emails to all applicants who got rejected (the employer must prewrite the rejection emails).

2. WriteQuestions Class

In this class the employer will write the questions and can choose between MCQ’s or handwritten questions or both. After writing the questions, the employer can add them to the interview.

WriteQuestions Methods

Method 5: writeMCQ
Description: In this method, the employer can write 1 or more MCQ and give all answers points or else the answers will be set to 1 point each (Logic error).

Method 6: writeHandWritten
Description: In this method, the employer can write 1 or more handwritten questions, the employer cannot give them points not until the applicant answers these questions, meaning the questions must be manually given points.

Method 7: addQuestions
Description: After the employer writes the questions (MCQ and/or handwritten) the employer will use this method to add the questions into a timeline. The questions are available to the applicants once the timeline begins and ends when the timeline is zero.

Method 8: addTimeline
Description: In this method, the employer can add the timeline of the whole hiring period and the timeline of answering the question.

3. ApplicantPage Class

In this class, the applicant can enter his/her page after logging in. After that, the applicant can edit their page, search for a company and apply for a job.

ApplicantPage Methods

Method 9: editProfile

Description: In this method, the applicant can change his/her information by accessing the applicantData class and changing it from there. For example: name, password, email etc.

Method 10: applyJob

Description: In this method, the applicant can start answering the questions in the specific timeline and ends by clicking submit or by the time running out.

Method 11: searchCompany

Description: In this method, the applicant can search for companies that are hiring via the search bar. Searching by company name, company address etc.

4. CompanyPage Class

In this class, the employer can enter his/her page after logging in. After that, the employer can edit their profile and write questions.

CompanyPage Methods

Method 12: editProfile

Description: In this method, the employer can change information about the company by accessing the comapnyData class and changing it from there. For example: email, password, company number etc.

Method 13: writeQuestions

Description: This method will take the employer to the WriteQuestions class which can let the employer write MCQ’s and/or handwritten questions, add the questions and set a timeline.

5. LoginPage Class

In this class, the user gets to login as an applicant or as a company, if the user chooses the applicant login then the user will go to the applicant login otherwise then the user will go to the company login.

LoginPage Methods

Method 14: applicantLogin

Description: In this method, the user will enter his/her applicant information, name and password. The system will then check if the name and password entered are in the ApplicantData class, if it is, then login successful, if not, then the user will have to retry or login as a company.

Method 15: companyLogin
Description: In this method, the user will enter his/her company information, company name and password. The system will then check if the name and password entered are in the companyData class, if it is, then login successful, if not, then the user will have to retry or login as an applicant.

6. Registration Class

In this class, the user will be able to register as an applicant or company. If the user registers as an applicant, then all his/her information will be stored in the ApplicantData class. If the user registers as a company, then all company's information will be stored in the CompanyData. If the user is already registered, then the user can just go to the login page.

Registration Methods

Method 16: insertApplicant
Description: This method allows the user to register as an applicant by entering the following: Email, password, phone number and full name. if none of the entered was already taken (other than the name) then the registration is complete, and the information is stored in the AppplicantData class.

Method 17: insertCompany
Description: This method allows the user to register as a company by entering the following: Email, password, company number, landline number, business address and company name. if none of the entered was already taken, then the registration is complete, and the information is stored in the CompanyData class.

Method 18: goToLogin
Description: If the user has already registered, then the user can go to the login page without registering.

7. ApplicantData Class

This class stores all registered applicants in a data structure. Its also used in comparing the entered data in the applicant login method to confirm the login.

8. CompanyData Class

This class stores all registered companies in a data structure. Its also used in comparing the entered data in the company login method to confirm the login.