MIDDLE EAST TECHNICAL UNIVERSITY
ENGINEERING FACULTY
DEPARTMENT OF COMPUTER ENGINEERING
INITIAL DESIGN REPORT
SUBJECT: ONLINECV
ASSISTANT: ALI ORKAN BAYER

“ISKOLIK” BY AIVA
Table of Contents

1. INTRODUCTION ................................................................................................................. 4
   1.1 Project Scope and Definition ....................................................................................... 4
   1.2 Project Scope ............................................................................................................... 5

2. SYSTEM MODULES ............................................................................................................ 6
   2.1 Overview ..................................................................................................................... 6
   2.2 Job seeker module ...................................................................................................... 7
   2.3 Employer module ....................................................................................................... 7
   2.4 Administrator module ............................................................................................... 8
   2.5 Operator module ....................................................................................................... 8
   2.6 External system interaction module ......................................................................... 9
   2.7 Database module ...................................................................................................... 9
   2.8 Authentication module ............................................................................................. 10

3. BEHAVIORAL MODELING ................................................................................................. 11
   3.1 Use Case Diagrams ................................................................................................... 11
   3.2 Use Case Scenarios ................................................................................................... 13

4. DATA MODELING .............................................................................................................. 23
   4.1 Entity-Relationship Diagrams .................................................................................. 23
   4.2 Data Descriptions ..................................................................................................... 26

5. CLASS DIAGRAMS ........................................................................................................... 36
   5.1 Job-seeker module ..................................................................................................... 37
   5.2 Employer module ....................................................................................................... 37
   5.3 Administrator module ............................................................................................... 38
   5.4 Operator module ....................................................................................................... 38
   5.5 External system interaction module ......................................................................... 39
   5.6 Database module ....................................................................................................... 39
   5.7 Authentication module ............................................................................................. 40
1. INTRODUCTION

In preparation of this report our main goal is to explain the design of our project “Iskolik” as clearly as possible. This report is conceived in the way lit by the requirement analysis report written priorly. In our design report the scope of the project, functional and nonfunctional requirements, survey about the technologies, use case diagrams, activity diagrams and ER diagram had been mentioned about. Addition to the stuff we had prepared before, class diagrams, sequence diagrams and modular specifications will be revealed. The detailed design of our system will be presented in the final design report.

1.1 Project Scope and Definition

In this project we aim to build “a gateway portal for HR companies & institutions to provide service to job seekers and employers”. In other words we want to develop a web portal to be the single entry point to both job seekers and employers.

Our Web portal will be above all the job-seeking web sites and the job seeker won’t be obliged to submit his/her CV to all these web sites. When he/she creates a CV in our web portal, it will be sent to all the job-seeking web sites and also he/she will be able to see all the job announcements from all of the job-seeking web sites. Also our portal will be very helpful to the employers because he/she will be able to see appropriate job-seekers from all over the job-seeking web sites and also he/she will be able to create a job announcement when he pay a certain amount of money. It is obvious that by means of the services that our system will provide both job-seekers and HR institutions will be relieved.
Our system will be in contact with some job-seeking websites which we agreed. Users in our web portal will not need to enter all of the job-seeking websites on the Internet, our system will have the capability to do it for them. Consequently, our system will pretend the job-seekers and employers from wasting their time on the web to seek for job, job-seeker or submit CV, job announcement etc. Our system will keep all of the users in our system in contact with many job-seeking websites.

1.2 Project Scope

Our system will have many capabilities for job-seeker, employer, operator and administrator. Here are general features that will take place in our portal.

- All of the users will be obliged to register firstly to our system. Our system will send an activation mail to the e-mail address of user. If the user is to activate his/her account, a profile as job-seeker or employer will be initiated in our system.

- All of the users will be obliged to login to our system to use the web-portal. When he/she enters the correct login data (username and password) to the website then he/she will be directed to his/her home page. All the information regarded to his/her profile will be displayed in this home page.

- If the user has the job-seeker profile in our system, the functionalities provided for that user will be as follows: Firstly, he/she will be able to create a CV then submit it to our system and the other job-seeking websites which supply services to us. Secondly, he/she will be able to search job opportunities, all of the relevant job opportunities from our system and the other job-seeking websites, which supply services to us, will be displayed. Finally he/she will be able see some job opportunities in his/her home page regarding to his/her sector, city etc. This notification system is supplied by our web portal if the user wants.
• If the user has the employer profile in our system, the functionalities provided for that user will be as follows: Firstly, he/she will be able to form job announcement. After he/she pay a certain amount of money, our system will submit it to one or many of other job-seeking (chosen by the employer) websites which supply services to us. Secondly, he/she will be able to search for the job-seekers, all of the relevant job-seekers’ CVs information from our system and the other job-seeking websites, which supply services to us, will be displayed. Finally he/she will be able see some job-seekers CVs in his/her home page regarding to job announcements submitted by him/her. This notification system is supplied by our web portal if the user wants.

• If the user has the administrator profile in our system, the functionalities provided for that user will be as follows: Firstly, he/she will be able to manage the users among the system. Administrator can warn some users if the user submits an inappropriate CV or job announcement, remove some users from the system. Secondly he/she will be able to remove some inappropriate job announcements from the system.

• There will be one more profile in our system which is operator. If the user has the operator profile in our system, the functionalities provided for that user will be as follows: The operator will be in charge of the money transactions and changing the payment status of the job announcement if necessary.

2. SYSTEM MODULES

2.1 Overview

Our system consists of seven modules, namely Database, Operator, External System, Authentication, Job-seeker, Employer and Administrator. During division of our system into modules our main goal was to create modules that would have least connection with other modules i.e. which would handle most operations within. We decided to make separation by logical functionalities that are required from our system like operator admin
and external systems. Two modules database and external system do interfacing to
database and external system, so that each of other modules do not need to handle
interfacing with those separately, we call these *interfacing modules*. Authentication
handles with registration and logging in existing users. Once logged in user’s operations
are handled by corresponding module. is redirected to corresponding profile. Other
modules correspond to different user profiles: job-seeker, administrator, operator and
employer.

### 2.2 Job Seeker Module

Job-seeker module handles all operations job-seeker might request are handled either
directly or indirectly by calling methods of interfacing modules. Classes with relations
among them are shown in corresponding class diagram in class diagrams section. CVInfo
class contains all data CV contains, CVManagement handles validation submission and
displaying operations of CV. For submission it calls corresponding external system and
database module routines. Job-seeker class encapsulates runtime information relevant to
the job-seeker. SearchJobWS is essentially web-service which implements job-seeking
functionality. The feedback received from external systems is send as input to
ProcessJobSearchFeedBackWS which sorts feedback. After sorting the feedback is
presented to job-seeker. Notification submodule consists of two webservices NotifyWS
and ProcessWS, NotifyWS is triggered by job-announcement posting, inside it makes
use of ProcessWS for determining which job-seekers to notify about that posting.

### 2.3 Employer Module

Employer module handles all operations employer might request are handled either
directly or indirectly (by calling methods of interfacing modules). Classes along with
relations among them are shown in corresponding class diagram in class diagrams
section. AnnouncementInfo class contains all data announcement contains.
AnnouncementManagement handles validation posting and displaying operations of
Announcement. For posting it calls corresponding external system and database module
routines. Employer class encapsulates runtime information relevant to the job-seeker.
SearchEmployeeWS is a web-service which implements employee-seeking functionality. The feedback received from external systems is send as input to ProcessEmployeeSearchFeedBackWS which sorts feedback. After sorting the feedback is presented to job-seeker. Notification sub-module consists of two web-services NotifyWS and ProcessWS, NotifyWS is triggered by CV-posting, inside it makes use of ProcessWS for determining which employees to notify about that posting. AcknowledgeOfPayment as its name implies handles the issue of acknowledging employee of payment associated with posting of an announcement.

2.4 Administrator Module

Administrator module handles all operations employer might request are handled either directly or indirectly (by calling methods of interfacing modules). Classes along with relations among them are shown in corresponding class diagram in class diagrams section.

- AddUser() - manual adding of a new user.
- ActivateUser() – activating user
- DeactivateUser() – deactivating user
- DeleteUser() – removal of user
- FetchAnnouncement() – fetching list of announcements
- FetchCV() fetching CVs of users
- DeleteAnnouncement() – removal of announcement
- CheckJobAnnouncement() – checking of job-announcement
- ChangeJobAnnouncementStatus() – change status of an announcement
- FetchUserActivities() – bring activities list of a user
- SendMessageToUser() – send email to user

2.5 Operator Module
Operator module handles all operations operator might request are handled either directly or indirectly (by calling methods of interfacing modules). Classes along with relations among them are shown in corresponding class diagram in class diagrams section.

- FetchTransactions() – fetch transactions list
- ApproveTransactions() – approve a transaction

2.6 External system interaction module

External System serves as a single interface to external systems, so that other modules do not keep information of how this interfacing is made. It handles issues of connecting logging in and messaging with external systems. Its web-services make use of external-systems’ web-services provided to our system. External systems are required to provide all web-services corresponding to web-services in this module.

- SubmitAnnouncementWS() – posts announcements to given external systems
- SubmitCVWS() – submits CV to external systems
- SearchJobAnnouncementWS() – searches external systems for announcements, with given parameters.
- SearchEmployeeWS() – searches for employees in external systems with parameters supplied to it.
- LoginWS() – logs to appropriate external system
- RegisterWS() – registers to appropriate external system
- ValidateWS() – checks whether user is registered to given external system

2.7 Database Module

Database Manager module provides a single interface to database, so that other modules do not keep handle database related operations directly, but instead call routines of database manager. It handles issues of connecting to and querying and modifying contents database,

- AddNewUser() – record newly registered user information.
- CheckLoginData() – checks if the login data is valid
• AddAnnouncement() – saves announcement
• AddCV() – saves CV
• RecordExtSysLoginData() – records external system authentication info. (this is necessary because in most cases to make use of external system web-services one should login to external system).
• ChangeUserStatus() – records user status modification
• AddTransaction() – add new transaction information to database
• RemoveTransaction() – erase transaction from database(usually automatic on accomplished transactions)
• ValidateWS() – checks whether authentication info of a particular user in particular external system exists.
• CheckJobAnnouncements() – checks job announcement
• ChangeJobAnnouncementStatus() – records change of status of a given announcement.
• DeleteAnnouncement() – remove given announcement from database
• FetchUserActions() – fetches list of actions performed by given user, which are recorded during call of AddCV() AddAnnouncement() etc.
• AddUserMessage() – adds a message to messagelist of particular user.
• UpdateUserInfo() – update user information
• RelevantUsers() – fetches list of users relevant to a particular user. Used in notification system, for instance, for getting list of employees to be notified by CV submission of a given job-seeker.
• AddEmployerNotification() – add a notification to a given employer
• AddJobSeekerNotification() – add a notification to a given job-seeker
• FetchTransactions()- fetches list of transactions
• ApproveTransaction()- approve given transaction

2.8 Authentication Module

Authentication module handles authentication operations, which are classified as logging and registering to the system. All operations are handled either directly or indirectly (by
calling methods of interfacing modules). Classes along with relations among them are shown in corresponding class diagram in class diagrams section.

- Login() – for logging to the system, user is then redirected to corresponding page corresponding to his profile type.
- Register() – for registration of users, applies to job-seekers and employees only. Administrators and operators are logged into the system through Login() WS, but are registered manually.

3. BEHAVIORAL MODELING

3.1 Use Case Diagrams
### 3.2 Use Case Scenarios

![Job Announcement Management Activity Diagram]

**Job Announcement Management Activity Diagram**
Observe User actions

Nothing that needs intervention

Ban relevant user

Send warning message to relevant user

Deliver the message to relevant user

Make the status of the user banned

User Management Activity Diagram
Submission of job announcement by the employer invokes this activity

Search for relevant job seekers

Generate current opportunities list appropriately

Job-seeker notification Activity Diagram
Employee Seeking Activity Diagram
Job Announcement Activity Diagram
Search for relevant employers

Generate potential employee lists appropriately

This activity is invoked by submission of CV of a relevant job seeker

Employer Notification Activity Diagram
Choose Criteria
(from State/Activ...)

Validity Check
(from State/Activ...)

[Valid]

Login to the relevant systems
(from State/Activ...)

Search the systems
(from State/Activ...)

Process the feedback
(from State/Activ...)

Present to the user
(from State/Activ...)

Job Seeking Activity Diagram
CV Submission Activity Diagram
Money Transaction Management Activity Diagram
User Login Activity Diagram
4. DATA MODELLING

4.1 Entity-Relationship Diagrams
CV
- CV_id
- Jobseeker_id
- Nationality
- Birth_date
- Birth_country
- Birth_city
- Father_name
- Mother_name
- Gender
- Marital_status
- Military_status
- Driving_License_Status1
- Driving_License_Status2
- Home_phone
- Cellular_phone
- Address
- Country
- City
- Town
- Post_code
- Hobbies
- Registered_clubs
- Blood_type
- Smoking
- Preferred_workstyle
- Preferred_city1
- Preferred_city2
- Preferred_city3
- Preferred_city4
- Preferred_country1
- Preferred_country2
- Preferred_country3
- Preferred_country4
- Preferred_sector_id1
- Preferred_sector_id2
- Preferred_sector_id3
- Preferred_sector_id4
- Preferred_position_id1
- Preferred_position_id2
- Preferred_position_id3
- Preferred_position_id4
- CV_active

Foreign Language
- Language_id
- CV_id
- Language
- Degree
- Writing_skill
- Reading_skill
- Speaking_skill
- Learned_place

School
- School_id
- CV_id
- School_name
- Department
- Country
- City
- Start_month
- Start_year
- Finish_month
- Finish_year
- Grading_system
- Graduate_grade

Reference
- Reference_id
- CV_id
- Name
- Surname
- Company
- Sector_id
- Position_id
- Phone_number
- Cel_phone
- Email
- Reference_type

Experience
- Experience_id
- CV_id
- Company_name
- Sector_id
- Position_id
- Organization_area
- Start_date
- Finish_date
- Work_type
- Country
- City

Computer Skill Job-seeker
- Skill_id
- CV_id
- Computer_skill
- Degree

 Desired Web-sites for Job-Seeker
- Des_website_id
- Website_id
- CV_id
- Website_active

Job Seeker
- Jobseeker_id
- User_name
- Password
- TC_kimlik_no
- Name
- Surname
- Email1
- Email2
- Security_Question
- Security_Question_Answer
- User_active

Web-Sites Job-Seeker
- Website_id
- Website_name
### 4.2 Data Descriptions

<table>
<thead>
<tr>
<th>Name</th>
<th>Job Seeker</th>
</tr>
</thead>
</table>
| **Where/How Used** | Registration page  
CV viewing page |
| **Description** | This table will hold personal information of users that registered for job-seeking |

#### Field Names:
- **Jobseeker_id** (Long Integer – Auto Incremented)
- **User_name** (VarChar(10)): username selected by user
- **Password** (VarChar(10)): password selected by user
- **TC_kimlik_no** (VarChar(11))
- **Name** (VarChar(20))
- **Surname** (VarChar(20))
- **Email1/Email2** (VarChar(50))
- **Security_Question** (VarChar(60))
- **Security_Question_Answer** (VarChar(60))
- **User_active** (Integer): Show whether user is activated by admin or not

<table>
<thead>
<tr>
<th>Name</th>
<th>CV</th>
</tr>
</thead>
</table>
| **Where/How Used** | CV viewing page  
CV form page |
| **Description** | This table will hold all information of a CV that filled by a job-seeker |

#### Field Names:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CV_id</td>
<td>(Long Integer – Auto Incremented)</td>
</tr>
<tr>
<td>Jobseeker_id</td>
<td>(Long Integer): Job-seeker’s id from job seeker table determines which user submitted related CV.</td>
</tr>
<tr>
<td>Nationalty</td>
<td>(VarChar(20))</td>
</tr>
<tr>
<td>Birth_date</td>
<td>(Date)</td>
</tr>
<tr>
<td>Birth_country</td>
<td>(VarChar(20))</td>
</tr>
<tr>
<td>Birth_city</td>
<td>(VarChar(20))</td>
</tr>
<tr>
<td>Father_name</td>
<td>(VarChar(20))</td>
</tr>
<tr>
<td>Mother_name</td>
<td>(VarChar(20))</td>
</tr>
<tr>
<td>Gender</td>
<td>(VarChar(1))</td>
</tr>
<tr>
<td>Marial_status</td>
<td>(VarChar(10))</td>
</tr>
<tr>
<td>Military_status</td>
<td>(VarChar(10))</td>
</tr>
<tr>
<td>Driving_Licence_Status1/2</td>
<td>(VarChar(10))</td>
</tr>
<tr>
<td>Home_phone</td>
<td>(VarChar(15))</td>
</tr>
<tr>
<td>Cellular_phone</td>
<td>(VarChar(15))</td>
</tr>
<tr>
<td>Address</td>
<td>(VarChar(100))</td>
</tr>
<tr>
<td>Country</td>
<td>(VarChar(20)): Country name where user reside</td>
</tr>
<tr>
<td>City</td>
<td>(VarChar(20)): City name where user reside</td>
</tr>
<tr>
<td>Town</td>
<td>(VarChar(30)): Town name where user reside</td>
</tr>
<tr>
<td>Post_code</td>
<td>(VarChar(10))</td>
</tr>
<tr>
<td>Hobbies</td>
<td>(Text): Hobbies of user</td>
</tr>
<tr>
<td>Registered_clubs</td>
<td>(Text): Social clubs which user is member of</td>
</tr>
<tr>
<td>Blood_type</td>
<td>(VarChar(10))</td>
</tr>
<tr>
<td>Smoking</td>
<td>(VarChar(10)): State whether user smokes or not</td>
</tr>
<tr>
<td>Preferred_workstyle</td>
<td>(VarChar(10)): Work style (part-time,full-time,etc.)</td>
</tr>
<tr>
<td>Preferred_city</td>
<td>(VarChar(20)): City where user prefers to work</td>
</tr>
<tr>
<td>Preferred_country</td>
<td>(VarChar(20)): Country where user prefers to work</td>
</tr>
</tbody>
</table>
**Preferred_sector_id** (Long Integer): Sector id from Sector table. It determines in which sector user prefers to work

**Preferred_position_id** (Long Integer): Position id from Position table. It determines which job position the job-seeker prefers (manager, engineer, etc.)

**CV_active** (Integer): State whether CV can be viewed by employers or not

<table>
<thead>
<tr>
<th><strong>Name</strong></th>
<th>School</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Where/How Used</strong></td>
<td>CV viewing page&lt;br&gt;CV form page</td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>This table will hold education information of Job-Seekers, table filled when forming CV</td>
</tr>
</tbody>
</table>

**Field Names:**

- **School_id** (Long Integer – Auto Incremented)
- **CV_id** (Long Integer): CV_id from CV table determines which CV includes related school information.
- **School name** (VarChar(30))
- **Department** (VarChar(30))
- **Country** (VarChar(20)): Country where school is located
- **City** (VarChar(20)): City where school is located
- **Start_month** (VarChar(15)): Start month of school
- **Start_year** (Integer): Start year of school
- **Finish_month** (VarChar(15)): Finish month of school
- **Finish_year** (Integer): Finish year of school
- **Grading_system** (Integer): Gradng system used in that school (5, 4, 10, etc.)
- **Graduate_grade** (Float Number)
<table>
<thead>
<tr>
<th>Name</th>
<th>Experience</th>
</tr>
</thead>
</table>
| Where/How Used                  | CV form page  
CV viewing page |
| Description                     | This table holds work experiences information of job-seekers which are filled while CV forming |

**Field name:**

- **Experience_id** (Long Integer – Auto Incremented)
- **CV_id** (Long Integer): CV id from CV table. It determines which CV includes related experience
- **Company_name** (VarChar(30))
- **Sector_id** (Long Integer): Sector id from Sectors table. It determines at which sector the related experience is.
- **Position_id** (Long Integer): Position id from Position table to show preferred position.
- **Start_date** (Date)
- **Finish_date** (Date)
- **Work_type** (VarChar(30)): Work type (part-time, full-time, etc.)
<table>
<thead>
<tr>
<th><strong>Country</strong> (VarChar(20))</th>
<th>Work place of work-experience (country)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>City</strong> (VarChar(20))</td>
<td>Work place of work-experience (city)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Name</strong></th>
<th><strong>Reference:</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Where/How Used</strong></td>
<td>CV form page</td>
</tr>
<tr>
<td></td>
<td>CV viewing page</td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>This table includes reference information of job-seekers in CV</td>
</tr>
</tbody>
</table>

**Field names:**
- **Reference_id** (Long Integer – Auto Incremented)
- **CV_id** (Long Integer): CV id from CV table. It determines which CV includes related reference
- **Name** (VarChar(30)): Name of reference
- **Surname** (VarChar(30)): Surname of reference
- **Company** (VarChar(30)): Company in which related reference work
- **Sector_id** (Long Integer): Sector id from Sector table. It determines the sector at which related reference work
- **Position_id** (Long Integer): Position id from Positions table. It determines the position of reference person in his/her company
- **Phone number** (VarChar(15)): Phone number of reference person
- **Cell phone** (VarChar(15)): Cellular phone number of reference person
- **Email** (VarChar(50)): E-mail address of reference person
- **Reference type** (VarChar(15)): Professional, personal, etc.

<table>
<thead>
<tr>
<th><strong>Name</strong></th>
<th><strong>Foreign language</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Where/How Used</strong></td>
<td>CV form page</td>
</tr>
<tr>
<td></td>
<td>CV viewing page</td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>Holds information about foreign language skills of job-seeker which is filled in CV form page</td>
</tr>
</tbody>
</table>
**Field names:**

*Language_id* (Long Integer – Auto Incremented)

*CV_id* (Long Integer): CV id from CV table. It determines which CV includes related reference

*Language* (VarChar(20)): Language name

*Writing skill* (Integer): Knowledge level of writing skills (1 to 10)

*Reading skill* (Integer): Knowledge level of reading skills (1 to 10)

*Speaking skill* (Integer): Knowledge level of speaking skills (1 to 10)

*Learned place* (VarChar(30)): Place where related language learned

---

<table>
<thead>
<tr>
<th>Name</th>
<th>Desired web-sites for job-seeker</th>
</tr>
</thead>
<tbody>
<tr>
<td>Where/How Used</td>
<td>CV form page</td>
</tr>
<tr>
<td></td>
<td>CV viewing page</td>
</tr>
<tr>
<td>Description</td>
<td>Job seekers will select web-sites that their CV will be sent and this table holds selected web-sites’ information.</td>
</tr>
</tbody>
</table>

**Field names:**

*Des_website_id* (Long Integer – Auto Incremented)

*Website_id* (Integer): Website id from Web-sites table. Selected web-sites’ information can be get from web-sites table by this id.

*CV_id* (Long Integer): CV id from CV table. It determines in which CV the related web-site is selected to send CV

*Website_Active* (Integer): It determines whether this web-site is activated or deactivated to send CV by administrator or user.

---

<table>
<thead>
<tr>
<th>Name</th>
<th>Web-sites for Job-seeker</th>
</tr>
</thead>
</table>
| Where/How Used | CV form page  
|               | CV viewing page  
|               | (It is necessary for “Desired Web-sites for Job-seekers” Table) |
| Description   | This table includes all external web-sites’ information.  
|               |  
|               | **Field names:**  
|               | Website_id (Integer – Auto Incremented)  
|               | Website_Name (VarChar(50)): Link of the related web-site |

<table>
<thead>
<tr>
<th>Name</th>
<th>Employer</th>
</tr>
</thead>
</table>
| Where/How Used | Registration page  
|               | Job Announcement Listing page  
|               | Job Announcement viewing page  
|               | Job Announcement form page |
| Description | This table will hold company and related person information of users that registered as Employer  
|             |  
|             | **Field Names:**  
|             | Employer_id (Long Integer – Auto Incremented)  
|             | Employer_active (Integer): It shows whether registration of this employer is approved or not.  
|             | User_name (VarChar(10)): username selected by user  
|             | Password (VarChar(10)): username selected by user  
|             | TC_kimlik_no (VarChar(11))  
|             | Name (VarChar(20)): Name of employer  
|             | Surname (VarChar(20)): Surname of employer  
|             | Email (VarChar(30)): E-mail address of employer  
|             | Phone (VarChar(15)): Phone number of employer  
|             | Fax (VarChar(15)): Fax number of employer  
<p>|             | Security_Question (VarChar(50)) |</p>
<table>
<thead>
<tr>
<th>Name</th>
<th>Announcement</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Where/How Used</strong></td>
<td>Job Announcement listing page</td>
</tr>
<tr>
<td></td>
<td>Job Announcement viewing page</td>
</tr>
<tr>
<td></td>
<td>Job Announcement form page</td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>This table hold announcement information of employers</td>
</tr>
<tr>
<td><strong>Field name:</strong></td>
<td></td>
</tr>
<tr>
<td><em>Announcement id</em> (Long Integer – Auto Incremented)</td>
<td></td>
</tr>
<tr>
<td><em>Announcement_active</em> (Integer): It shows whether that announcement can be viewed in announcement searches. It will change by preference of employer.</td>
<td></td>
</tr>
<tr>
<td><em>Employer_id</em> (Long Integer): Employer_id attribute from Employer table. It determines which employer send related announcement.</td>
<td></td>
</tr>
<tr>
<td><em>Announcement title</em> (VarChar(20))</td>
<td></td>
</tr>
<tr>
<td><em>City</em> (VarChar(20)): Working Place (City)</td>
<td></td>
</tr>
<tr>
<td><em>Country</em> (VarChar(20)): Working Place (Country)</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Computer_skill_employer</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>Where/How Used</td>
<td>Job announcement form page</td>
</tr>
<tr>
<td></td>
<td>Job announcement viewing page</td>
</tr>
<tr>
<td>Description</td>
<td>This table holds wanted computer skills by employer in an announcement.</td>
</tr>
</tbody>
</table>

**Field names:**

- **Computer_skill_id** (Long Integer – Auto Incremented)
- **Announcement_id** (Long Integer): Related announcement’s id from announcement table.
- **Computer_skill** (VarChar(30)): Name of computer skill
- **Degree** (Integer): Knowledge level of computer skill

<table>
<thead>
<tr>
<th>Name</th>
<th>Desired web-sites for Employer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Where/How Used</td>
<td>Job announcement form page</td>
</tr>
<tr>
<td></td>
<td>Job announcement viewing page</td>
</tr>
<tr>
<td>Description</td>
<td>This table holds selected web-sites by employer to send his/her announcement, this table is important for payment of selected web-sites.</td>
</tr>
</tbody>
</table>
**Field names:**

*Des_website_id* (Long Integer – Auto Incremented)

*Website_id* (Long Integer): Web-site id from Website table, for getting information of related web-site

*Employer_id* (Long Integer): Employer_id attribute from Employer table. It determines which employer selects the related web-site.

*Website_active* (Integer): it shows whether employer pay for the announcement cost of related web-site

<table>
<thead>
<tr>
<th>Name</th>
<th>Web-sites for Employer</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Where/How Used</strong></td>
<td>Job Announcement form page</td>
</tr>
<tr>
<td></td>
<td>Job Announcement viewing page</td>
</tr>
<tr>
<td></td>
<td>(It is necessary for “Desired Web-sites for Employer” Table)</td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>This table includes all external web-sites’ information.</td>
</tr>
</tbody>
</table>

**Field names:**

*Website_id* (Long Integer – Auto Incremented)

*Website_Name* (VarChar(50)): Link of the related web-site

<table>
<thead>
<tr>
<th>Name</th>
<th>Sector</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Where/How Used</strong></td>
<td>CV form page</td>
</tr>
<tr>
<td></td>
<td>CV viewing page</td>
</tr>
<tr>
<td></td>
<td>Job Announcement form page</td>
</tr>
<tr>
<td></td>
<td>Job Announcement viewing page</td>
</tr>
<tr>
<td></td>
<td>(It is necessary for CV, Experience, Reference, Employer and Announcement tables)</td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>This table includes all necessary sectors’ list.</td>
</tr>
</tbody>
</table>

**Field names:**

*Sector_id* (Long Integer – Auto Incremented)
### 5. CLASS DIAGRAMS

Components of our system and their dependencies

![Class Diagram](image-url)
5.1 Job Seeker Module

Class diagram for Job-Seeker Module

 CVManagement
  GetData()
  Check()
  Submit()

 JobCriteria

 SearchJobWS
  doSearch()

 Job-seeker

 CVInfo
  String name
  String surname
  Date birthDate
  long id
  String workarea
  int experience
  String[] languagesSpoken
  String name2

 Job-seeker Notification
  NotifyWS()
  ProcessWS()

 ProcessJobSearchFeedbackWS
  ProcessJobSearchFeedback()

 User
  long id
  String ip
  String[] session

5.2 Employer Module

Class diagram for Employer Module

 User

 AnnouncementManagement
  GetData()
  CheckFields()
  Submit()

 EmployeeCriteria
  Create()

 Employer

 AnnouncementInfo

 AcknowledgeEmployerOfPayment

 EmployerNotification
  NotifyWS()
  ProcessWS()

 SearchEmployeeWS
  doSearch()

 ProcessEmployeeSearchFeedbackWS
  ProcessEmployeeSearchFeedback()
5.3 Administrator Module

Class diagram for Administrator Module

- Admin
  - adminName : String
  - password : String
  - AddUser()
  - ActivateUser()
  - DeactivateUser()
  - DeleteUser()
  - FetchAnnouncements()
  - FetchCV()
  - DeleteAnnouncement()
  - CheckJobAnnouncement()
  - ChangeJobAnnouncementStatus()
  - FetchUserActivities()
  - SendMessageToUser()

5.4 Operator Module

Class diagram for Operator Module

- User
  - opname()
  - Login()
  - Register()

- OperatorModule
  - FetchTransactions()
  - ApproveTransaction()
5.5 External system interaction module

Class diagram for External System Module

<table>
<thead>
<tr>
<th>ExternalSystem</th>
</tr>
</thead>
<tbody>
<tr>
<td>SubmitAnnouncementWS()</td>
</tr>
<tr>
<td>SubmitCVWS()</td>
</tr>
<tr>
<td>SearchJobAnnouncementWS()</td>
</tr>
<tr>
<td>SearchEmployeeWS()</td>
</tr>
<tr>
<td>LoginWS()</td>
</tr>
<tr>
<td>RegisterWS()</td>
</tr>
<tr>
<td>ValidateWS()</td>
</tr>
</tbody>
</table>

5.6 Database Module

Class diagram for Database Module

<table>
<thead>
<tr>
<th>DatabaseManager</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection con</td>
</tr>
<tr>
<td>AddNewUser()</td>
</tr>
<tr>
<td>CheckLoginData()</td>
</tr>
<tr>
<td>AddAnnouncement()</td>
</tr>
<tr>
<td>AddCV()</td>
</tr>
<tr>
<td>RecordExtSysLoginData()</td>
</tr>
<tr>
<td>ChangeUserStatus()</td>
</tr>
<tr>
<td>AddTransaction()</td>
</tr>
<tr>
<td>RemoveTransaction()</td>
</tr>
<tr>
<td>ValidateWS()</td>
</tr>
<tr>
<td>CheckJobAnnouncements()</td>
</tr>
<tr>
<td>ChangeJobAnnouncementStatus()</td>
</tr>
<tr>
<td>DeleteAnnouncement()</td>
</tr>
<tr>
<td>FetchUserActions()</td>
</tr>
<tr>
<td>AddUserMessage()</td>
</tr>
<tr>
<td>UpdateUserInfo()</td>
</tr>
<tr>
<td>RelevantUsers()</td>
</tr>
<tr>
<td>AddEmployerNotification()</td>
</tr>
<tr>
<td>AddJobSeekerNotification()</td>
</tr>
<tr>
<td>FetchTransactions()</td>
</tr>
<tr>
<td>ApproveTransaction()</td>
</tr>
</tbody>
</table>
5.7 Authentication Module

Class diagram for Authentication Module

```plaintext
LoginWS
- String userName
- String password
- Bool job-seeker
- Login()

RegisterWS
- String userName
- String password
- Bool job-seeker
- Register()
```
6. SEQUENCE DIAGRAMS

Sequence Diagram for Announcement Posting

1: Create
2: New AnnouncementInfo instance
3: CheckFields
4: true if Announcement valid
5: 
6: AddAnnouncement
7: ValidateWS
8: Authentication info
9: RegisterWS
10: UpdateUserInfo
11: LoginWS
12: SubmitAnnouncementWS
13: Submission confirmation
14: Submission confirmation
Sequence Diagram for CV Submission

Job Seeker User Interface

: CVInfo

1: Create()
2: New CVInfo instance

: CVManagement

3: Check( )
4: true if CV valid

: DatabaseManager

5: Submit( )
6: AddCV()

: ExternalSystem

7: ValidateWS( )
8: Authentication info
9: RegisterWS( )
10: UpdateUserInfo( )
11: LoginWS( )
12: SubmitCVWS( )
13: Submission confirmation
14: Submission confirmation
Sequence Diagram for Employee Seeking

Employer User Interface

EmployeeCriteria

1: Create()

2: new criteria instance

3: doSearch()

4: ValidateWS()

5: true if user is registered, false otherwise

6: RegisterWS()

7: Authentication info

8: UpdateUserInfo()

9: LoginWS()

10: SearchEmployeeWS()

11: Rough Employee List

12: ProcessEmployeeSearchFeedback()

13: Processed Employee List

ExternalSystem

DatabaseManager

SearchEmployeeWS

ProcessEmployeeSearchFeedback
Sequence Diagram for Employer Notification

1: Submit( )

2: RelevantUsers( )

3: Relevant Users List

4: AddEmployerNotification( )

Job Seeker Interface

: CVManagement

: DatabaseManager
Sequence Diagram for Job Announcement Management

Admin User Interface

1: FetchAnnouncements( )

: Admin

2: CheckJobAnnouncements( )

: DatabaseManager

3: Announcement List

4: Announcement List Table

5: ChangeJobAnnouncementStatus( )

6: ChangeJobAnnouncementStatus( )

7: DeleteAnnouncement( )

8: DeleteAnnouncement( )
Sequence Diagram for Job Seeking:

1. Create()
2. Job Criteria instance
3. doSearch()
4. ValidateWS()
5. true if user is registered, false otherwise
6. RegisterWS()
7. Authentication info
8. UpdateUserInfo()
9. LoginWS()
10. SearchJobAnnouncementWS()
11. Rough Job List
12. ProcessJobSearchFeedback()
13. Processed Job Announcement List
Sequence Diagram for Job Seeker Notification System

Employer Interface

: AnnouncementManagement

1:

Database...

2: RelevantUsers( )

3: Relevant Users List

4: AddJobSeekerNotification( )

Sequence Diagram for Login Functionality

Generic User Interface

: User

1: Login( )

2: CheckLoginData( )

3: Success on correct data

4: Redirect user to Appropriate page

: DatabaseManager
Sequence Diagram for Money Transaction Management Sequence Diagram

1: FetchTransactions( )
2: FetchTransactions( )
3: Transactions List
4: Transaction List
5: ApproveTransaction( )
6: ApproveTransaction( )

Sequence Diagram for Registration Functionality

1: Register( )
2: AddNewUser( )
3: Confirmation
4: Confirmation
Sequence Diagram for User Management

Admin User Interface

1: FetchUserActivities()

2: User Actions Table

Admin

3: FetchUserActions()

4: Actions List

: Admin

5: DeactivateUser()

6: ChangeUserStatus()

: DatabaseManager

7: ActivateUser()

8: ChangeUserStatus()

9: SendMessageToUser()

10: AddUserMessage()
7. TECHNICAL ISSUES

7.1. Technological Research

To clarify how to develop each functionality we needed to make research about the possible technologies that can be used. We probed into details of ESB, Web 2.0, AJAX, Web Services, JAVA, XML, UML, SOA, SOAP, JDBC and WSDL.

We will use JAVA as programming language in that it provides many useful facilities for web services, which play vital role in the implementation of our project.

For sake of better understanding and communication we used UML. It is a standardized specification convention for object modeling. We used this convention to reveal requirements, behavioral model and structural design of our system. Yet we have modeled the requirements by forming the use-case diagrams, the behavioral model and structural design by means of activity diagrams, class diagrams and sequence diagrams.

Web Services have significant importance in our project. Web services use XML to decode the data and SOAP document to be transferred using open protocols. Using web services our application can publish its function or message as web-applications. Basically we will use web services to use the services provided by other websites for us and publish our services to other websites.

XML (Extensible Mark-up Language) is designed to store, carry, and exchange data but it is not designed to display data. XML is the most important technology in our project in that communication and data transfer between our system and the job seeking websites will be via XML.

An enterprise service bus (ESB) is a pattern of middleware that unifies and connects services, applications and resources within a business. It's a way of integrating applications, coordinating resources and manipulating information. ESB is very important for our project because we will make use of this technology when we use the
Web services of other job-seeking websites. Also ESB improves operational performance and reduces costs while simplifying the task of connecting dissimilar applications across a network of different operating systems.

Web 2.0 refers to a perceived second generation of web-based communities. It contains AJAX, XML, CSS, XHTML. We will also use Web 2.0 technology in our project. It contains especially AJAX and XML. By means of AJAX with an HTTP request, a web page can make a request to and get a response from a web server - without reloading the page. The user will stay on the same page without noticing that scripts request pages or sending data to a server in the background. We will use this approach in our project because loading the user requested data becomes faster and easier.

SOA (Service Oriented Architecture) represents a model in which functionality is decomposed into small, services, which can be distributed over a network regardless of the operating systems and can be combined together and reused to create business applications. We have adopt this architecture in the scope of our project.

WSDL (Web Services Description Language) is an XML-based language for describing Web services and how to access them. The document describes a Web service. It specifies the location of the service and the operations (or methods) the service exposes. We need this language to form our system’s web service implementation and build the web service architecture.

SOAP (Simple Object Access Protocol) provides a way to communicate between applications running on different operating systems, with different technologies and programming languages. We need to include SOAP documents in some steps of our project. The most important reason that we need SOAP is to exchange information between applications over HTTP. HTTP is supported by all Internet browsers and servers so SOAP was created to accomplish this. In many steps of our project we will need to send information to other applications and services in other websites and take information from them. SOAP provides us the ability to manage this situation.
7.2. Security of the system

Security is one of most important concerns in design and implementation of our system. Below we explain how we plan to deal with security concerns.

First of all data sent over network will be encrypted. We plan to use SSL protocol for this. The SSL protocol protects data from alteration and disclosure while it is in transit at the same time it ensures that users are connecting to web sites they think they connect.

Another measure we plan to take is stateless user authentication. We will give user a random session id which will be stored on both client and server sides, this will be renewed each 5-10 minutes. Id will consist of a long string of characters. With SSL session id is not exposed.

To prevent from breaking user password by corresponding breaking programs, we plan to do as gmail. The idea is that once user name and password is incorrectly entered several times, we will require entering not only username and password information but also the string which will be displayed on login window.

At the same time we will ensure that there is an administrator having many privileges over the system always monitoring the system.
8. USER INTERFACE

8.1 Employer Pages

8.1.1 Employer Registration

Employer Registration window is a part of user interface for employer registration. On the right part of the Register window, there are text edit boxes that have to be filled with correct information. Labels which are on just left of text boxes are used for defining the
needed information that must be filled in text box on the right. After clicking “Register” button, if information is valid all information will be recorded in database and registration will be completed successfully.

8.1.2 Employer Login

![Login Box](image)

Login Box has very simple structure that is formed by a button and two text boxes which have to be filled correctly to login. When the button is clicked, correctness of information is checked and if information is correct, user is transferred to profile page or other personal pages.

8.1.3 Employer Profile
On the left part of the Employer profile window, there are two operation boxes, namely Employer operations box and Search box. Search box will be explained in detail in 8.1.3.2. In Employer Operations box there are six links to other pages.

- **Main Page**: when clicked, main page of İşkolik will be opened.
- **My Profile**: now this link is not active, because the user is already on profile window.
- **Form/Update Job Announcement**: when clicked, job announcement for page is openeded.
• List My Job Announcements: when clicked, user’s job announcements list page will be opened, so user can view all of his/her announcements and if he wants he can modify them.
• Search for CVs: when clicked, search window will be opened.
• Notifications: when clicked, Notifications window will be opened.

On the right part of the Employer profile window, all profile information is displayed. On the bottom of window there is an “Update” button, for profile updating.

8.1.3.1 Employer Profile - Announcement Posting

On the left part of the Announcement Posting window, there are two bars, namely Announcement Form bar and Search bar. Search bar will be explained in detail in 8.1.3.2. In Announcement Form bar there are four links.
• Profile: when clicked, Employer Profile window will be opened..
• Form Announcement: this link is not active now, because the user is already in it.
• Desired Computer Skills: when clicked, Desired Computer Skills window is opened. In that window, user can fill computer skills that are requested from employee. That window is also a part of announcement form.
• Desired Web-Sites: when clicked, Desired Web-Sites window is opened. In that window, user can view and update his/her desired web-sites list.

On the right part of the Announcement Posting window, there are text boxes which must be filled correctly to form a correct job announcement. After filling all necessary parts of announcement form, user can save announcement by clicking “Save” button located at the bottom of the window.

8.1.3.2 Employer Profile - Employee Search

Employee Search Box has very simple structure. It consists of six text boxes which must be filled with search information and a button. When the button is clicked, search will be performed and results will be displayed on Search List window.
8.2 Job Seeker Pages

8.2.1 Job Seeker Registration

Job-Seeker Registration:

User Name
Password
Password (Again)
Identity Card Number
Name
Surname
E-mail 1
E-mail 2
Security Question
Answer of Security Question

Registration Window is for entering “Job Seeker” registration information. On the right part of the Register window, there are text edit boxes that have to be filled with proper information. Labels which are on the left of text boxes explain information to be entered to the corresponding textbox. After clicking “Register” button, if correct registration will be completed.

8.2.2 Job Seeker Login

Login Box has very simple structure that is formed by a button and two text boxes which have to be filled correctly to login. When the button is clicked, correctness of information
is checked and if information is correct, user is transferred to profile page or other personal pages.

8.2.3 Job Seeker Profile

On the left part of the Job-Seeker profile window, there are two operation boxes, namely Employer operations box and Search box. Search box will be explained in detail in 8.2.3.2. In Employer Operations box there are five links:

- Main Page: when clicked, main page of İşkolik is opened.
- My Profile: now this link is not active, because the user is already on profile window.
- Form/Update CV: when clicked, CV form page is opened.
- Search for Announcements: when clicked, search window is opened.
- Notifications: when clicked, Notifications window is opened.
On the right part of the Job-Seeker profile window, profile information is displayed. On the bottom of that window there is a button named “Update”, when it is clicked “Profile Update” window will be opened.

8.2.3.1 Job Seeker Profile - CV Posting
On the left part of the CV Posting window, there are two bars, namely CV Form bar and Search bar. Search bar will be explained in detail in 8.2.3.2 . In CV Form bar there are eight links.

- **Profile**: when clicked, Employer Profile window will be opened.
- **CV Form**: this link is not activated now, because CV Form window is already opened.
- **Schools**: when clicked, Schools window is opened. In that window, user can fill all of his/her school information. That window is also a part of CV form.
- **Foreign Languages**: when clicked, Foreign Languages window is opened. In that window, user can fill all of his/her foreign language information. That window is also a part of CV form.
• Experiences : when clicked, Experiences window is opened. In that window, user can fill all of his/her experience information. That window is also a part of CV form.
• References : when clicked, References window is opened. In that window, user can fill all of his/her reference information. That window is also a part of CV form.
• Computer Skills : when clicked, Computer Skills window is opened. In that window, user can fill all of his/her computer skill information. That window is also a part of CV form.
• Desired Web-Sites : when clicked, Desired Web-Sites window is opened. In that window, user can view his/her desired web-sites list and update the list.

On the right part of the CV Posting window, there are text boxes which must be filled properly to form a correct CV. After filling all necessary parts of CV form, user can save his/her CV by clicking “Save” button located at the bottom of the window.

8.2.3.2 Job Seeker Profile - Job Searching
Job Searching Box has very simple structure that is formed by six text boxes containing search specifications. When the button is clicked, search will be made and results will be listed in Search List window.

9. GANTT CHART
Responsibility Distribution Coloring:

<table>
<thead>
<tr>
<th>Blue: All Members</th>
<th>Pink: Fatma</th>
<th>Green: Emrah</th>
<th>Purple: Damir &amp; Emrah</th>
<th>Orange: Fatma &amp; Derya</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red: Damir</td>
<td>Yellow: Derya</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>