**Chapter Four**

**System design**

**4.1 Overview**

This system design document is for any kind of job activities that are available in Ethiopia. This document includes the design goal, the proposed system and object design.

**4.2 Purpose of the system**

This document describes the design issues of the overall system. It provides the complete architectural overview of the proposed system. It is intended to capture and express the significant architectural decisions, which have been made, on the system as well as to obtain the information needed to manage all citizens of the country.

**4.3 Design goal**

The Design Goals specify the qualities of the system that should be achieved and addressed during the design of the system. The design goals for the system are grouped into four categories. These are:

In this document, the design goals describe the qualities of the system that are derived from the non-functional requirements that can lead to decisions of developers.

The designing part is very important to make the implementation or development of the system through coding very simple. The different types of the system modeling techniques that are used for the implementation or development of the system such as deployment and component modeling are show in detail.

From the non-functional requirement of the system that was mentioned in SRS document, we can find the goal of our system.

 Performance

 Dependability

 Maintenance

 End user

**4.3.1 Performance**

**Response time: -** Depending on the network connection that the user machine has the system is going to interact and respond to user’s request in a maximum of a second, if the user is just viewing the pages, but if the user’s request requires the processing of the data base, like searching for schedule, is going to take an average of 1-5 seconds of communication latency with the server system. Moreover, streaming of trailers is going to have a response time up to 2-4 seconds.

**Memory:-**The client system requires an average of 10-15 megabits of RAM memory to be loaded on a user’s web browser and streaming of trailers require additional memory up to 20 megabits. The server system is going to require up to 40 GB of memory to store all the data and other components of the system.

**4.3.2 Dependability**

F-jobs application system should achieve the following dependability characteristics in order to resist crash and be available and reliable.

**Robustness:** -

**4.3.3 Maintenance**

In time of failure or need modification the system, need to be maintained. To be maintainable the system should meet the following maintenance criteria

**Extensibility: -** if it is needed to add new functionality to the system, this must be achieved by only making a separate page and integrate this page with the existing system.

**Modifiability: -** if in the system, some functionality requires to be modified, this modification must be done specifically to that function or page without affecting the overall system organization.

**4.3.4 End user**

From the user point of view, the system should provide the following end user criteria’s so that the system can achieve At least 90% usability by the user.

4.4 Proposed system  
4.4.1 Overview

This proposed system is mainly based on the SRS document that has already developed and  
submitted. It mainly deals with subsystem decomposition – dividing the system in to  
manageable components. Another major task in system design deals with hardware/software  
mapping, which deals with which components would be part in which hardware. Yet another  
aspect of system design deals with persistent data management, which illustrate as to how  
persistent data (file, database, etc.) are stored and managed and at last, Access control will be  
presented. Our system is three tires architectural it has **client side**, **server** and **database.**

Client side**:**

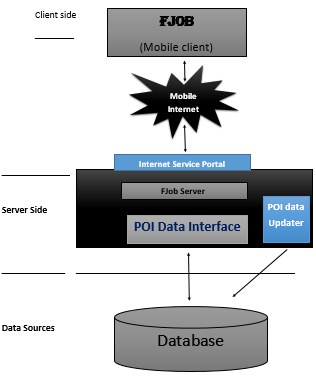
 Here in the client side Employee, Administrator, and user interface will be existing.

Figure 4.1 system design

Server side: Here the firebase servers to connect the data base application are found; that means the application logic to perform the application by the client is found.

Database**:** Here the databases that store the information are found.

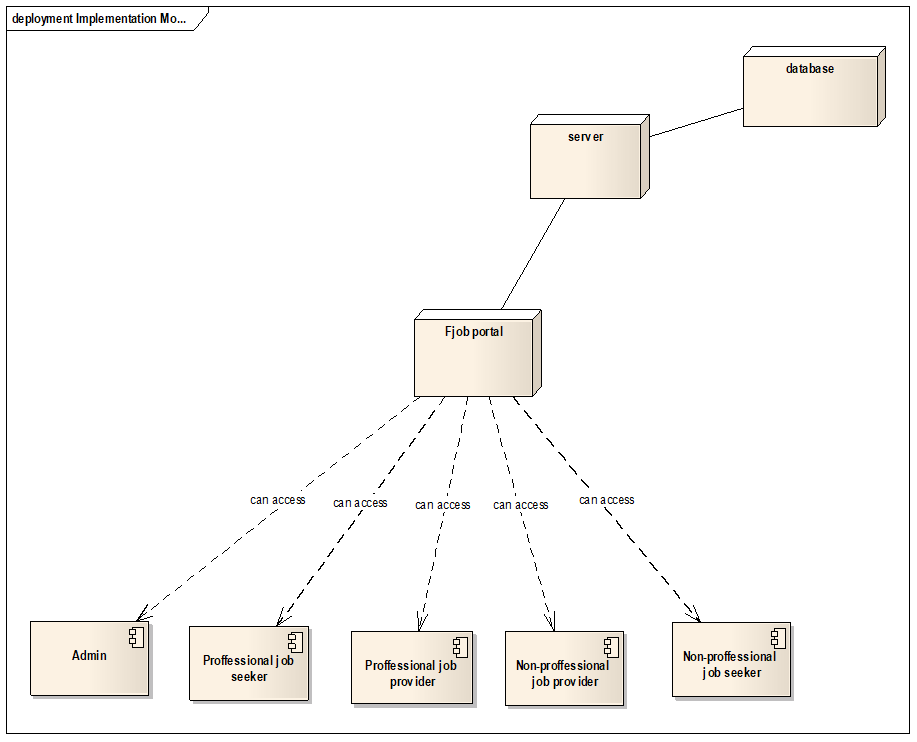
4.4.2 System architecture

Figure 4.2-system architecture of F-Job

**4.4.2.1 Internet service portal**an internet service portal is an initial entry to F-Job services.

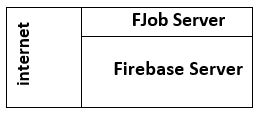


Figure 4.3 server overview

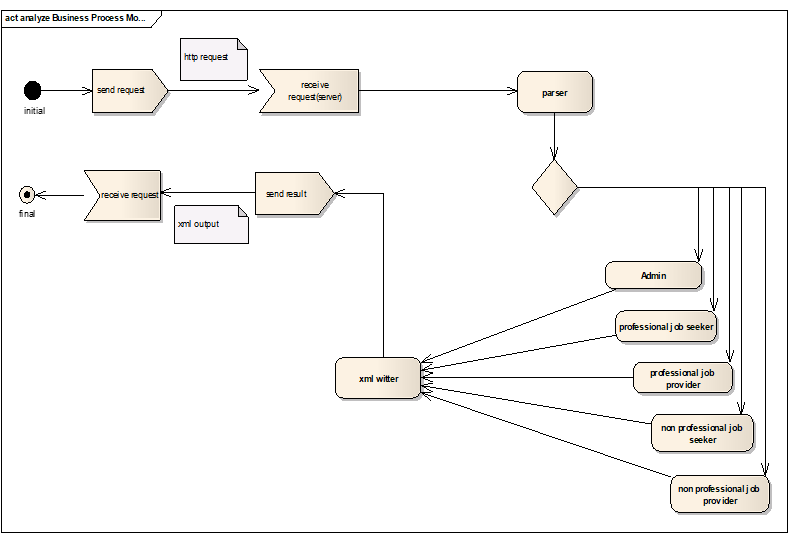
4.4.3 System process

Figure 4.4-system activity diagram

AdminThis section explores in detail about admin system architecture design. A top-level overview of the system, a brief overview of system processes or functionality are described in the below section.

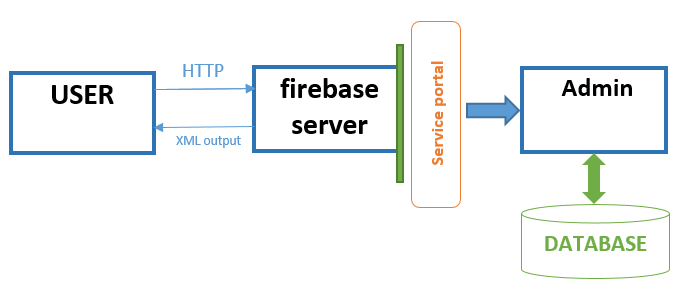


Figure 4.5 – admin system overview

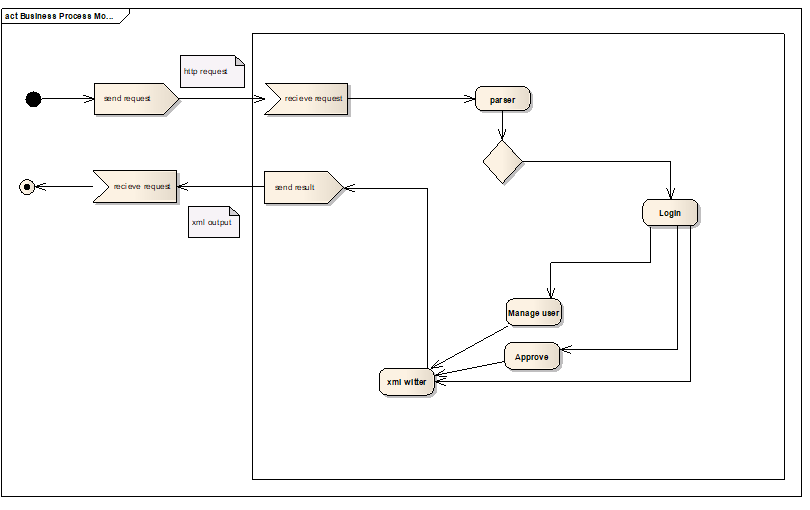
The figure shows admin system overview in relation to other components.

Figure 4.6-admin system process

Professional Job seeker

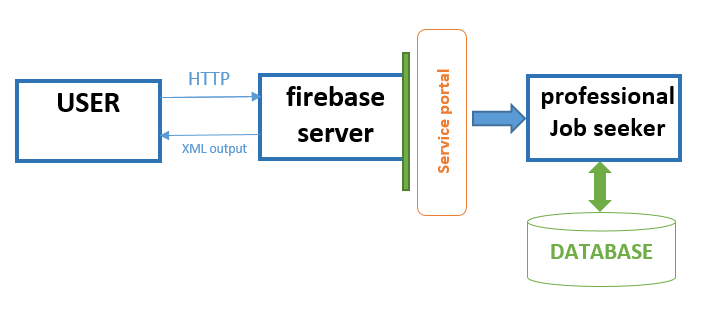


Figure 4.7 Professional Job seeker system overview

The figure shows Professional Job seeker system overview in relation to other components.

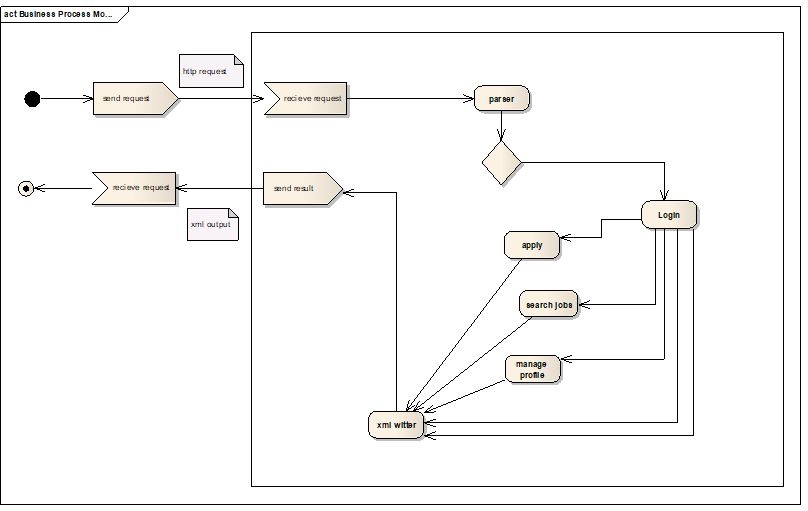
**Professional Job provider**

Figure 4.8 Professional Job seeker system process

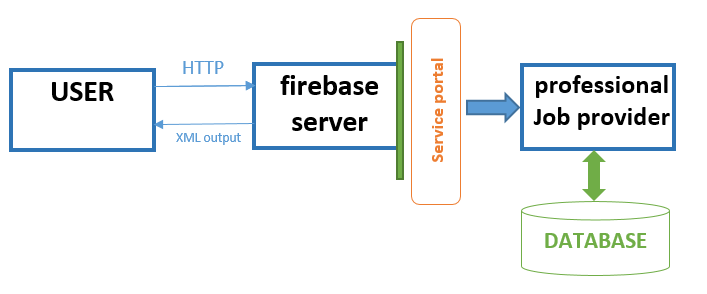


Figure 4.9 professional job-provider system overview

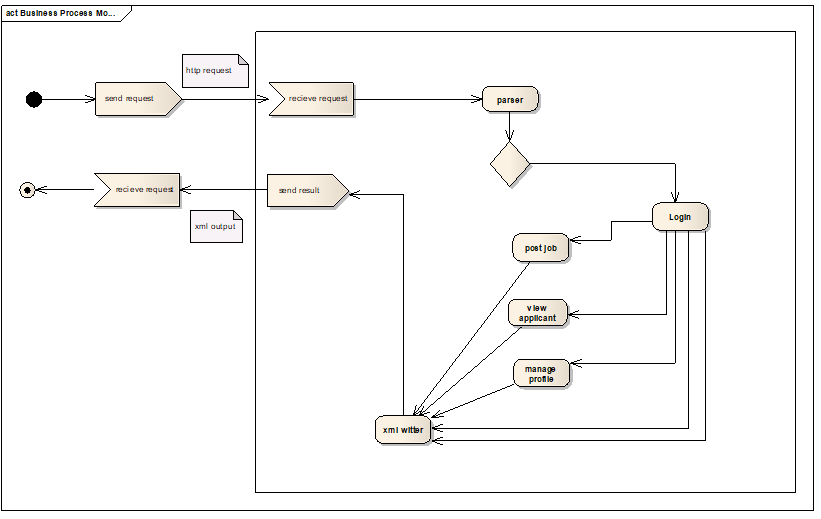
The figure shows professional job provider system overview in relation to other components

Figure 4.10 professional job provider system process

Non-professional job provider

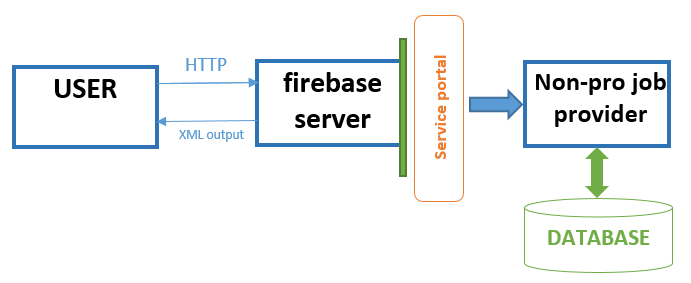
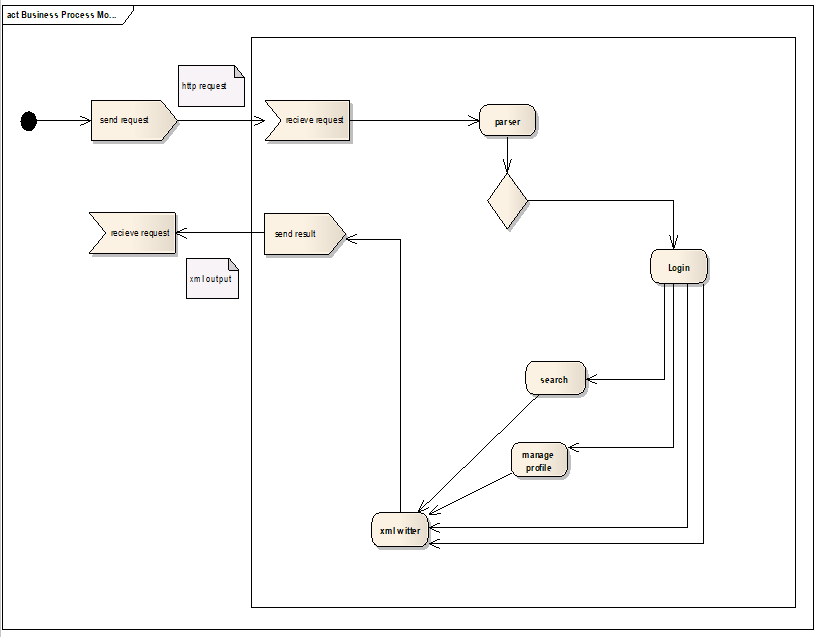


Figure 4.11 Non-professional job provider system overview

The figure shows professional job provider system overview in relation to other components

Figure 4.12 professional job-provider system process 

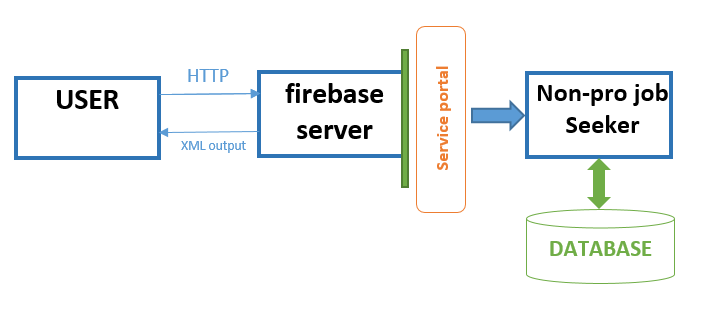
Non-professional job seeker

Figure 4.13 Non-professional job seeker system overview

The figure shows Non-professional job seeker overview in relation to other components

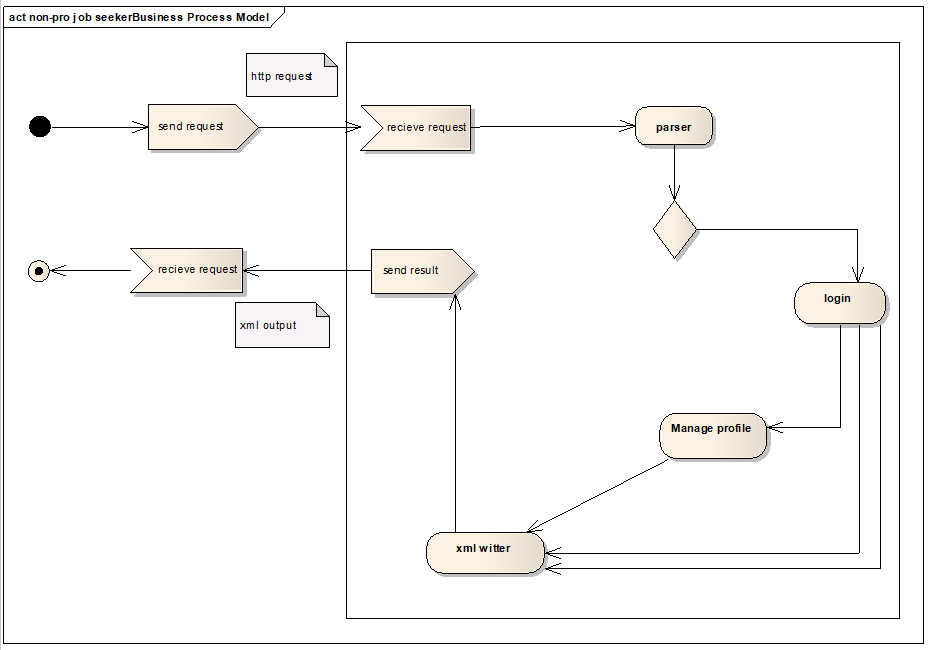


Figure 4.14 Non-professional job seeker system process

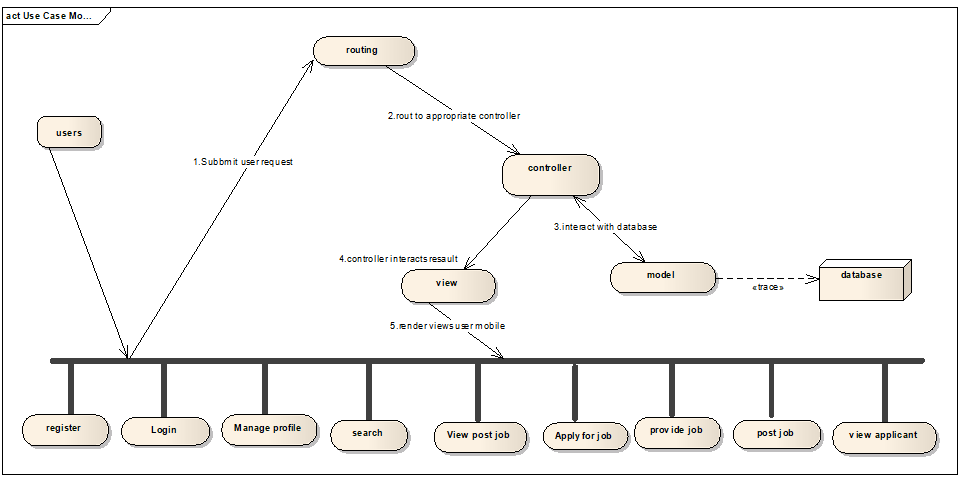
**4.4.4 How users are access our service**

Figure 4.15- user access system

4.4.5 Sub system decomposition

Subsystem decompositions will help to reduce the complexity of the system. The subsystems can be considered as packages holding related classes or objects. The CIS under consideration is decomposed in to subsystems as shown by the following diagram. These subsystems are further decomposed into other subsystems.

4.4.5.1 Deployment view

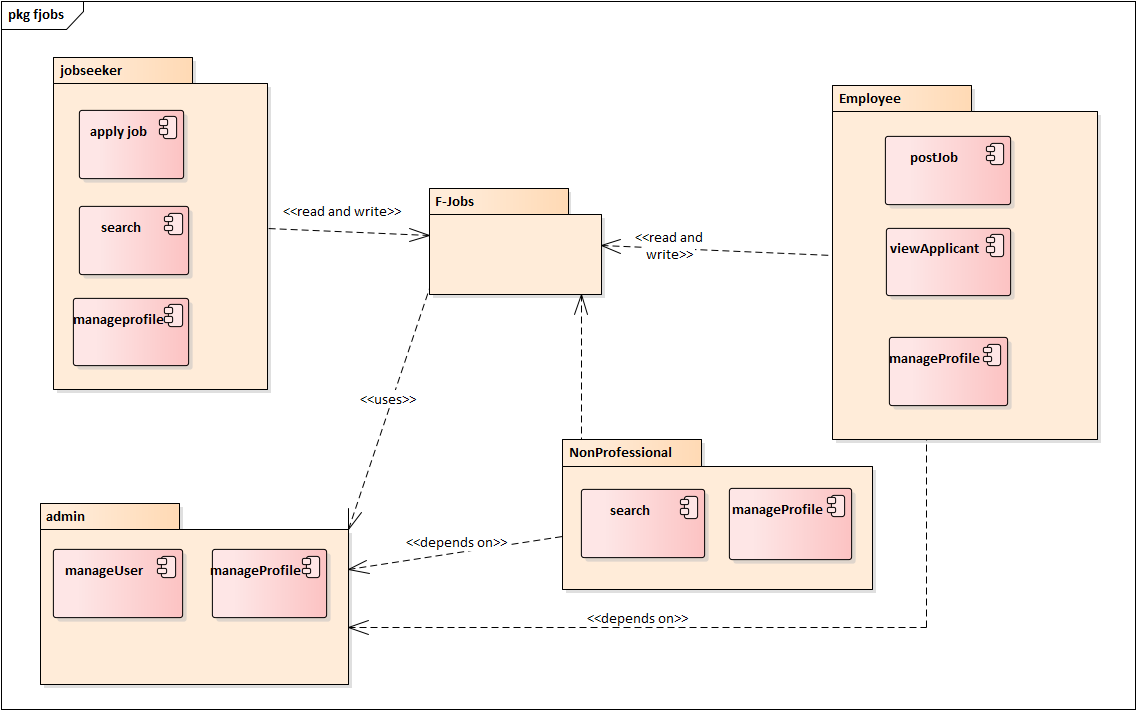


Figure 4.16- Deployment view

4.4.5.2 Sub system description

|  |  |  |
| --- | --- | --- |
| Admin | Responsible to provide an interface for manage user and manage profile | Menu  Nonprofessional seeker  Employer |
| Nonprofessional | Responsible to provide an interface for search and manage profile | Nonprofessional seeker  Menu |
| Jobseeker | Responsible to provide an interface for search job, Apply job and manage their profile | Menu  Application  Job List |
| Employee | Responsible to provide an interface for post job, view Applicant and manage profile | Job List  Jobseeker  Application  Menu |

4.1 Table of subsystem decomposition

4.4.6 hardware/software mapping

The system will have two processes, deployed in single or separate machine, that run in parallel, Namely, web server process and the database process. The database process, which runs on Server database engine, is responsible for maintaining storing an updated map information. Where us the web server process is responsible to host a restless web service that accepts a Request from the mobile application and send the updated information from the database server.

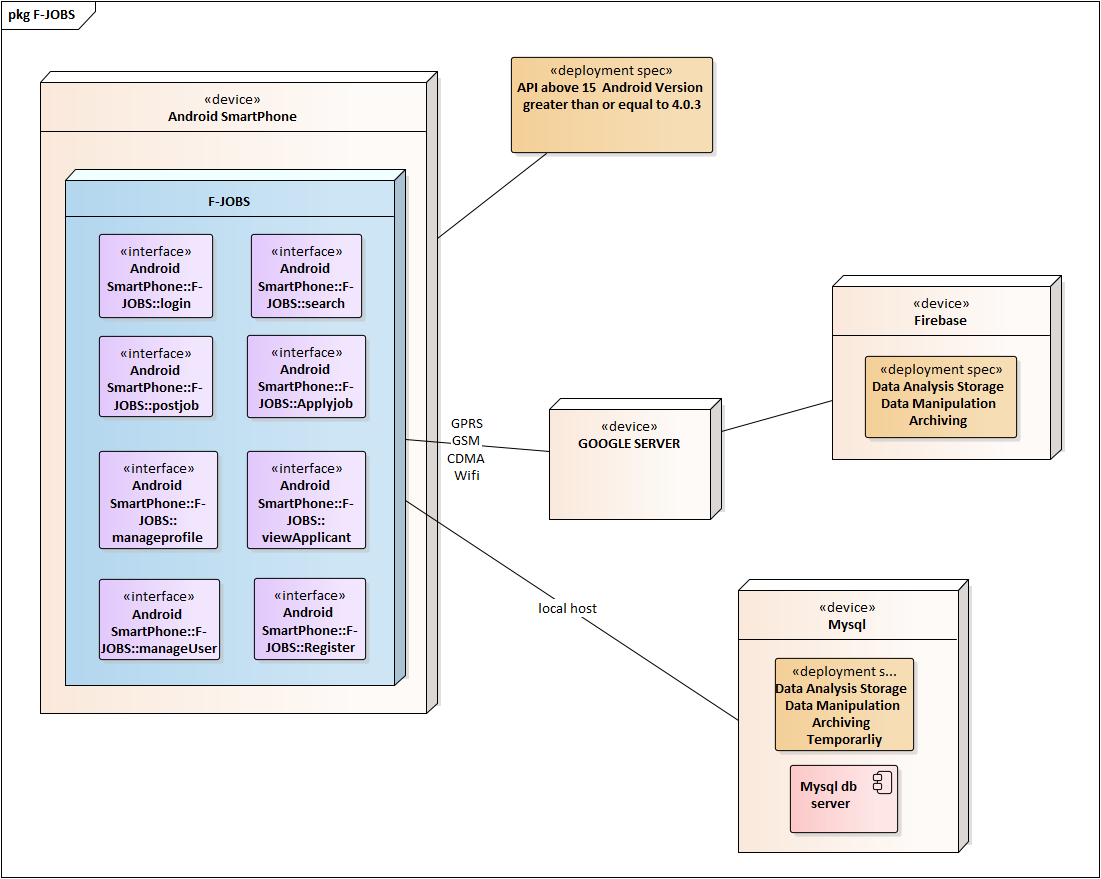


Figure 4.16- hardware/software mapping

4.4.7 Persistent data management

The purpose of this section is to show the mapping of the objects/classes of the system, identified during the analysis stage, in to the corresponding relational database.

Key

Mapped

Relational Database table

Class

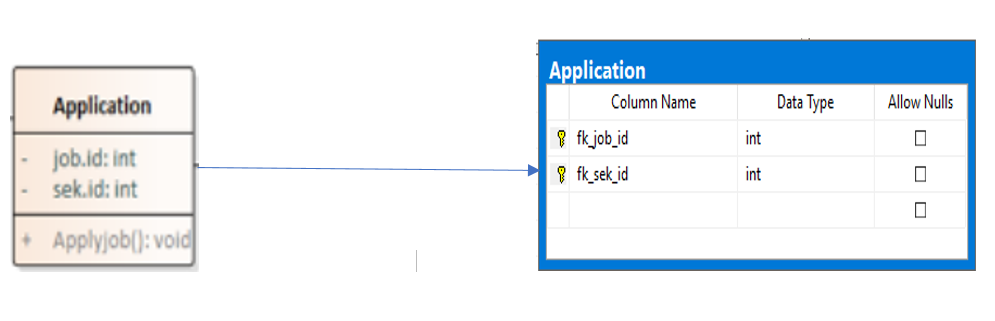


Figure 4.17 application class-database mapping

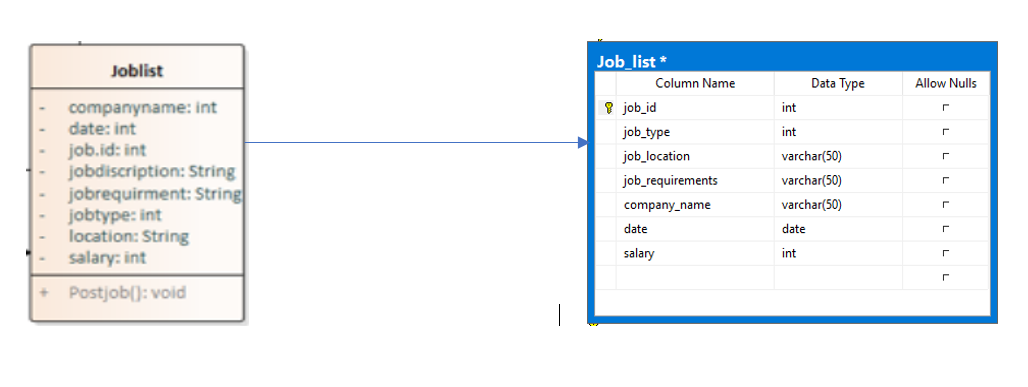


Figure 4.18 joblist class-database mapping

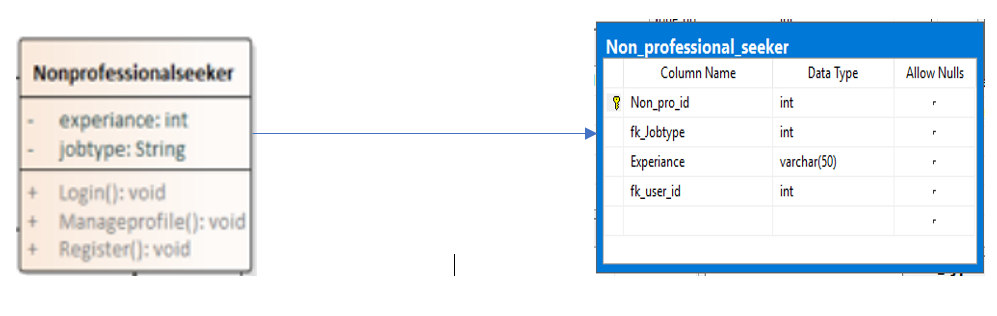


Figure 4.19 nonprofestional job seekers mapping

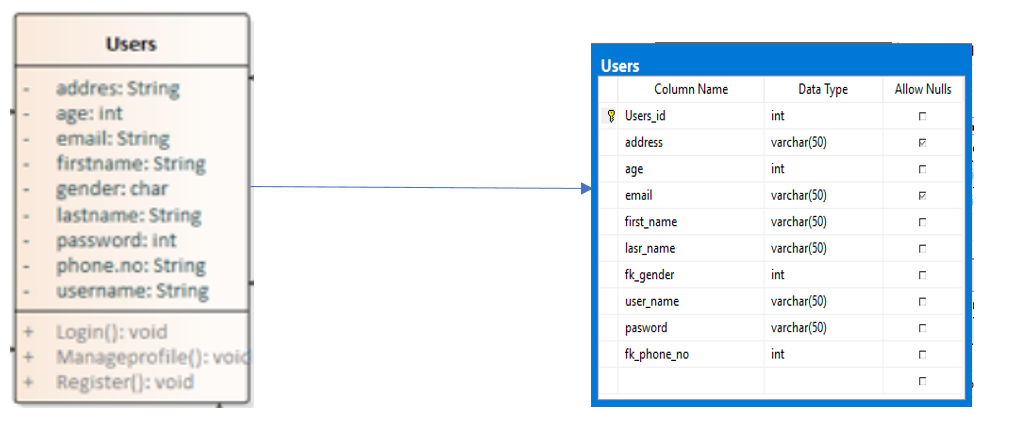


Figure 4.20 users class-databse mapping

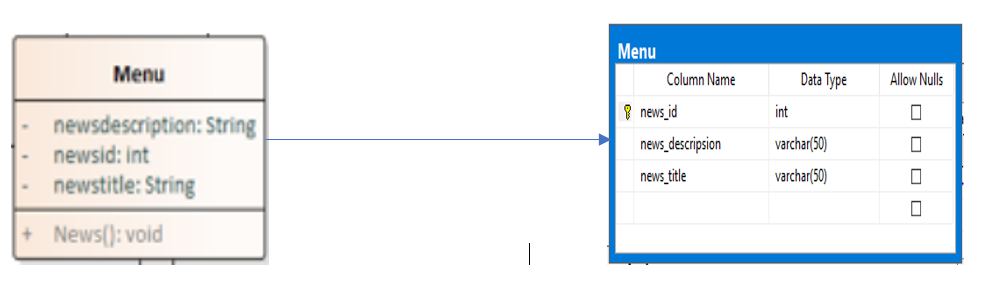


Figure 4.21 Menu mapping

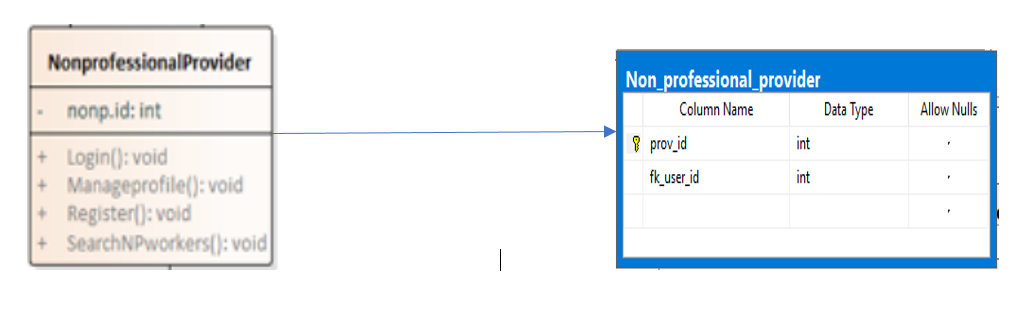


Figure 4.22 non professtional Employer mapping

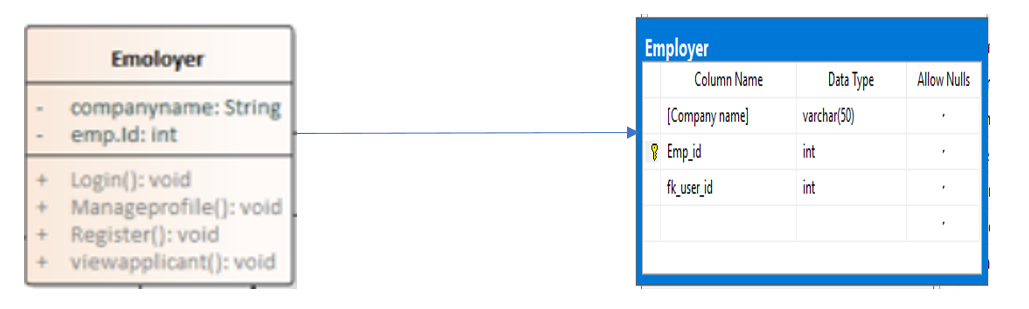


Figure 4.23 Employer class-databse mapping

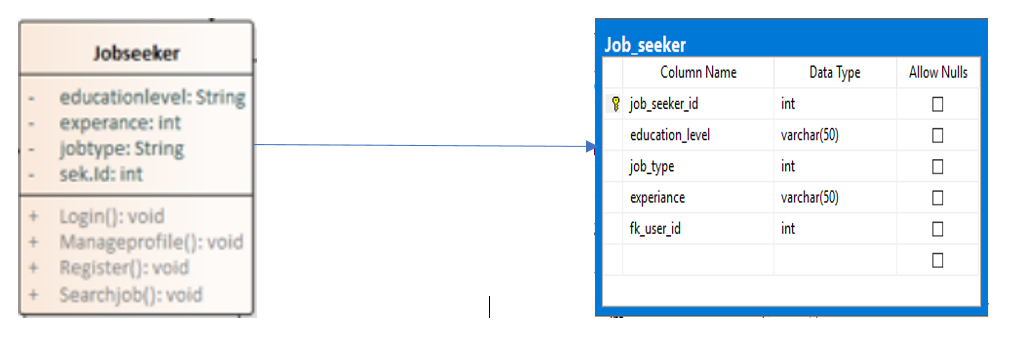


Figure 4.24 jobseeker class-databse mapping

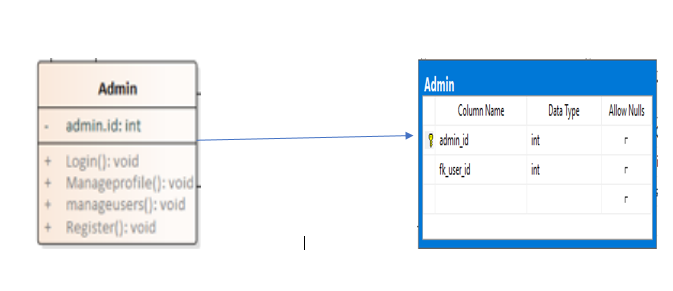


Figure 4.25 Admin class-databse mapping

4.4.8 Database design

In this part of the system design document the database is provided. The class in the class diagram as well as the relationship between them are converted in this portion. Generally, the database system is the backend part of any systems that helps the full functionality of the system.

The figure bellow shows the relationship mapping of each table in a relational database.

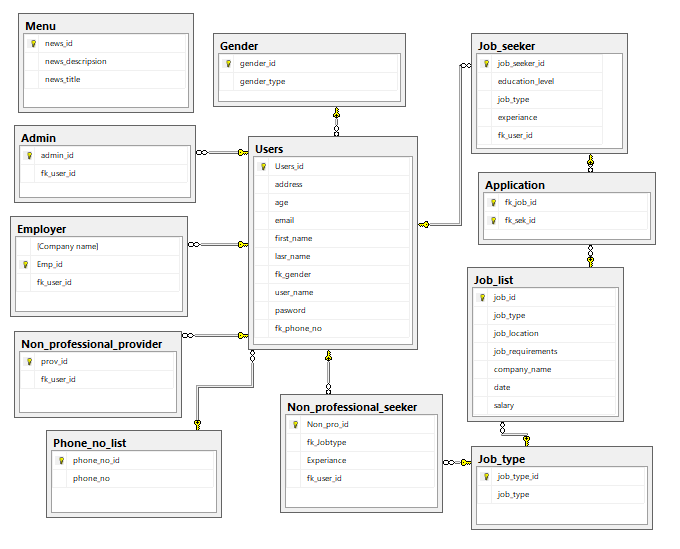


Figure 4.26 databse design

### 4.4.9 Component diagram

It is Describes all components in a system, their interrelationships, interactions, and the interface of the system. It is an outline of the composition structure of components or modules

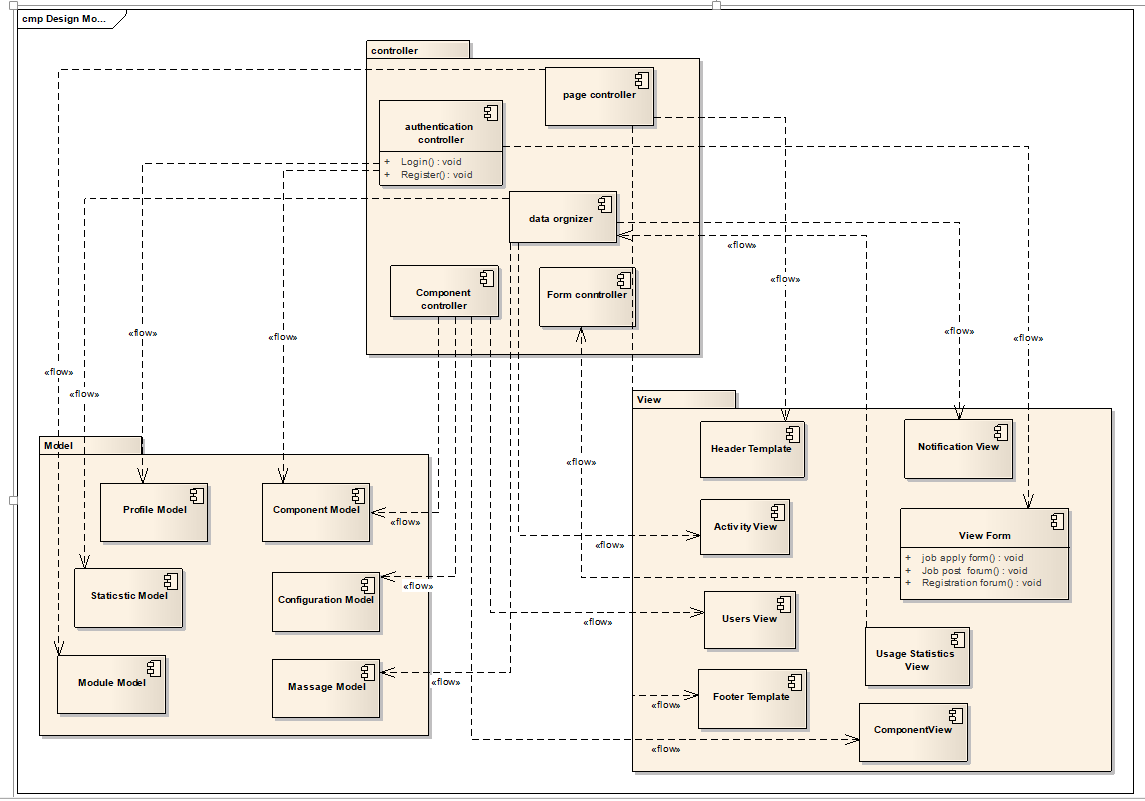


Figure 4.27 Admin class-databse mapping

**Component Description**

**Authentication controller**

**Description:** This is a class responsible to authenticate a user

**Role:** it is responsible to check if the validity of a user trying to login to the system

Page controller

**Description:** This is a route class, which will route to each pages as needed

**Role:** it is responsible to route each page requests in the system.

**Data organizer**

**Description:** the base component, which is directly linked to the users of the system and sends it back to the view called view activity.

**Role:** is responsible for tracking session and displaying the appropriate dash board and log activities of a user.

**Component controller**

**Description:** this is a class which extends the base controller class responsible to look for components.

**Role:** this class is responsible to control all the components in the system.

Form controller

**Description:** The base form controller class in which governs the form submission in the system

And sends back to the view called view form.

**Role:** this class is responsible for controlling of form submission before further process.

Profile model

**Description:** this is a class, which is linked with the profile table in the database.

**Role:** responsible for the extraction of record from and to the table

**Component model**

**Description:** this is a class which is linked with the database table in which used to determine the

Component to display to the user.

**Role:** responsible for the getting record from the database in order to display the appropriate

Component to the user.

**Statistics model**

**Description:** this is a class which will be connected to the statistics table in the database and

Responsible to send information to the specified data organizer controller.

**Role:** will fetch data from the database table and will pass it to the specified controller.

**Message model**

**Description:** this is a class, which will help to connect to the table in the database, and will help in the data organizer class.

**Role:** this will help to connect to the message table in the database and will retrieve record from

The table for further processing.

**Module model**

**Description:** this is a class, which is connected with the module table in the database and linked with the pages controller for further process.

**Role:** this class will be used to retrieve info from and to the module table and deliver to the pages Controller class.

**Configuration model**

**Description:** it is a class which is used to connect he component controller and the database table

**For data exchange.**

**Role:** is responsible to get access to the database table and return the data to the component controller for further process.

**Header template**

**Description:** this is the basic tinplating engine, which is the laravel blade

**Role:** this is used to customize the code so that it would be easier to debug

**Navigation template**

**Description:** this is the html file used to create the common navigation bar

**Role:** is used to create a common navigation bar

**Activity form**

**Description:** the html page that is related to the activity form layouts in the page

**Role:** this the view class, which is used to view the activity forms, fetched from the data organizer controller

**View form**

**Description:** this is the form viewer class using blade template and connected to the form controller class

**Role:** is used to create the basic form views that will be displayed to the user

**Notification view**

**Description:** this is the view class related to the notification views and connected to data organizer

**Controller class**

**Role:** used to view the notifications for the user

**Usage statistics view**

**Description:** this is the view class used to preview the statistical records to the user connected to data organizer class

**Role:** is used to the view to the user

**User view**

**Description:** this is the view class using the blade template engine connected to the component controller class

**Role:** used to create the user view to the user

**Footer template**

**Description:** the view class using the base template and connected to the footer controller class

**Role:** is used to view the footer to the user

**Component view**

**Description:** the view class using the blade template engine and connected to the component controller

**Role:** used to create the component view to the user

**4.4.10 Deployment diagram**

The following deployment diagram shows the F-JOBS. In the first node, which is the Users smartphone with the second node, F-JOBS pre-installed. The system contains the following interfaces inside it:

* Log In
* Post Job
* Search
* Apply job
* Manage profile
* Manage user
* Register
* View Applicant

The third node show the Google server is used to connect us with firebase. The fourth node is Firebase is where our data analysis, store, manipulate and archived. Moreover, the fifth node is MySQL that needs to data analysis, store, manipulate and archived temporarily.

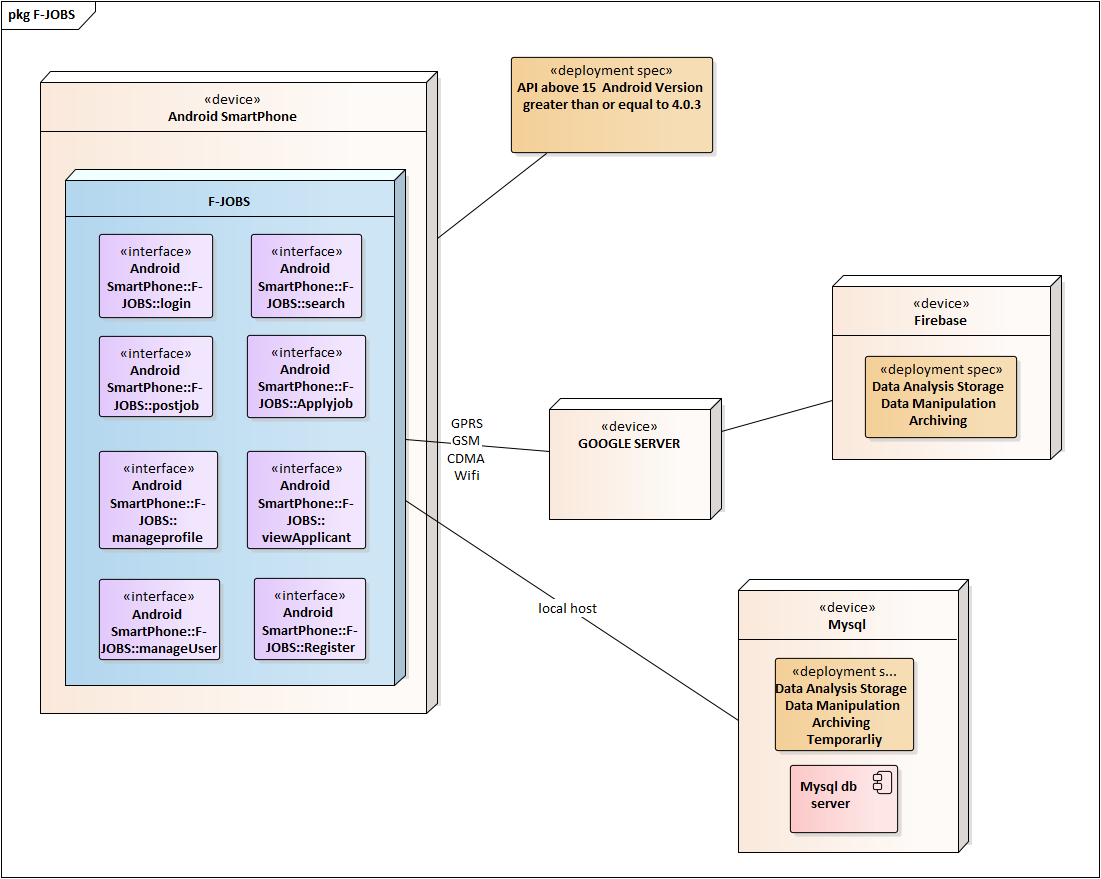


Figure 4.28 deploymentdiagram

### 4.4.11 Boundary condition

Administration use cases will be used in condition assuring of initiation and termination stages.

Dynamic Model of the System Startup

**THE INITIATION USE CASE:** The F-jobs system backend side must always be on the run for normal operation. Therefore, users of the application (system) must start the application from their android smart phone when they want to use the system. After the application (system) started up, it is also necessary to connect the Internet to access our server.

For administrator to be able to access to the backend system it is also necessary to connect the Internet through web browser or directly from the server computer, thus browsers must be installed on the non-server machines appropriate connection methods to connect to the server.

User Interface at Startup

User interface is Active only when the application was initiated and running thus is deactivated of on standby mode, when user is not interacting with the system.

Termination

**THE TERMINATION USE CASE:** After the user finishes his/her interaction with the application (system), this use case can be executes when user flushes ram standby processes from the running tasks screen

**Failure:** Possible things that may go wrong. Design foresees fatal failures (“failure use cases”).

|  |  |  |
| --- | --- | --- |
| **Error** | **Cause** | **Forecast Solution** |
| Insufficient Space on Device | Android allocates fixedstorage space for applications. Users cannot expand it. So user might run into this error even if his/her device indicates a great deal of remaining storage capacity. | Free up space |
| System User Interface (UI) not working | Could be many internal causes | restarting the device wiping the System UI cache |
| Some fatal errors | Malware | Scan with malware scanner |
| Internet not working | Improperly configured APN (access point name) settings. No internet from the user internet service provider (ISP).  Improper IP configuration  Proxy error | Wi-Fi resetting  Proper IP configuring  Configure APN to get mobile data. If these settings are not properly configured, user must manually set them. If the first step does not work user should contact his/her ISP. |
| Server Not found | Internet connection fail | Make sure Network connection or firewall (proxy) protection settings are correct. |

4.2 Table Design Goal Priority

**4.4.12 Access control**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Object actors | menu | user | job list | application |
| Employer | view news() | Register()  Login()  Manageprofile()  View applicant() | Post jobs() |  |
| Job seeker | view news() | Register()  Login()  Manageprofile() | search jobs() | Apply job() |
| Nonprofessional job provider | view news() | Register()  Login()  Manageprofile()  Searchnonprofestional seekers() |  |  |
| Nonprofessional job seeker | view news() | Register()  Login()  Manageprofile() |  |  |
| Admin | Post news() | Register()  Login()  Manageprofile()  Mangeusrs() |  |  |

4.2 access controller