

Freelancing Management System

FMS

The Team — Yaser, Umar, Taha and Boda

March 25, 2017

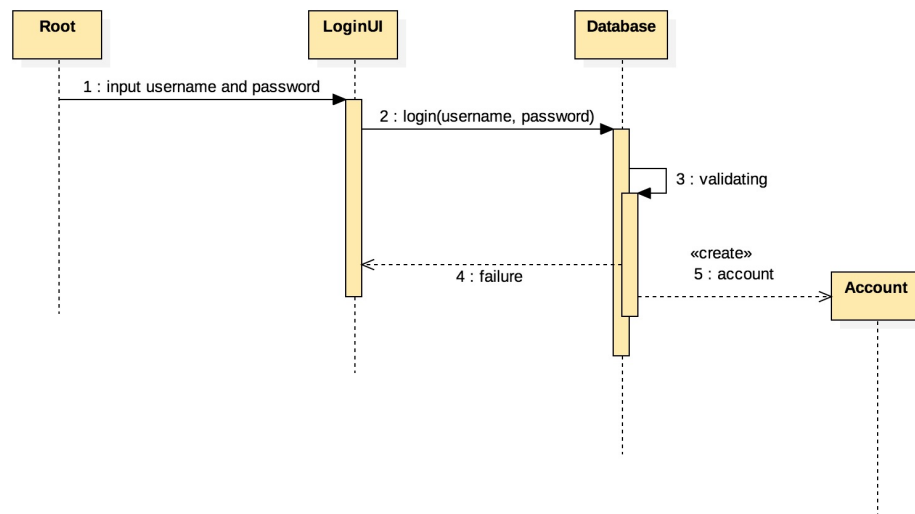


Figure 1: System Sequence Diagram of the *Root*

Contents

1	System Description	3
2	System Actors	3
2.1	Root	3
2.2	Admin	3
2.3	Freelancer	3
2.4	Employer	3
3	Diagrams	4
4	Use Case – <i>in detail</i>	4
4.1	Access Root	4
4.2	Specify Penalties	4
4.3	Add Admin	4
4.4	Remove Admin	5
4.5	Block Admin	5
4.6	Unblock Admin	5
4.7	Search Freelancers	6
4.8	Search about Freelancer	7
4.9	Review Employer history	8
4.10	Reject offer submitting	9
4.11	Accept Offer	10
4.12	Accept submitting Offer	11
5	Design Patterns	12
5.1	Singleton Design Pattern	12
6	Call Us	12

1 System Description

This is a *Freelancing Management System*, a system which allows *Customers* and *Freelancers* to find one-another easily. It manages the processes from finding workers, verifying them, engaging, paying up to rating. Let us take you on an interesting tour into the *system actors*, every one of them sees the system differently according to their role.

2 System Actors

2.1 Root

The Root is the owner of the system. He is the one who set it up for the first time with a special root access from him. The Root also specifies the system settings like penalties, bonuses on the rate and ratio of the profit on each Employer-Freelancer deal. Besides, He add *Admins* to manage the system.

2.2 Admin

Admins are the managers of the system. They have *The managers access* over the hole system, they can *delete* or *ban* any consumer account. Also, They can view some Statistics like the number of the Employers and Freelancer. They can *Receive* and *Reply Complaints* of the consumers as well. Admins can view reports about the system i.e *economic, consumers, blocked, ...* reports.

2.3 Freelancer

Freelancers are the workers of the system. Every one them have an *Profile* where he can share his experience, and work examples. His *Profile* contains enough information so that the Employer can decide wherever he is going to invest in that *freelancer* or not. There is a *Rate* in the *Freelancers Profile* is computed with very accurate algorithms based on the *Employers Feedback* and how commitment is the Freelancer! *Freelancer* are paid per hour, and it is in the *profile*. A *Freelancer* can review the *Employers' Profiles* and their *Tasks* which they can apply for it and make special offers for the *Employers*.

2.4 Employer

Employer are the stockholders. They also have their *profiles*. An *Employer* can create a *Task* or a *list of tasks* seen by interested *Freelancers*. Then the *Employer* can choose the best *Freelancer* among who applied. Once the *Employer* accepts the *offer* the money is *withdrawn* from his account to the pending state in the system. After the *task* is finished, the *Employer* can *rate* and *feedback* the *freelancer*, and sure the freelancer is paid.

3 Diagrams

4 Use Case – *in detail*

4.1 Access Root

Use Case Name	<i>Root Access</i>
Actors	<i>Root — Exclusively</i>
Pre-condition	<i>1. At configuration time for the very first use. 2. Root is given a default password.</i>
Basic Flow	<i>1. Open the System 2. Enter ‘root’ in the user name text box 3. Enter ‘0000’ in the password text box</i>
Post-condition	<i>have full control of FMS settings</i>

4.2 Specify Penalties

Use Case Name	<i>Specify Penalties</i>
Actors	<i>Root — Exclusively</i>
Pre-condition	<i>Root Access</i>
Basic Flow	<i>1. Open the system with the Root Access 2. Click on Specifies Penalties 3. Fill the Data 4. Click submit</i>
Post-condition	<i>Penalties is configurted</i>

4.3 Add Admin

Use Case Name	<i>Add Admin</i>
Actors	<i>Root — Exclusively</i>
Pre-condition	<i>Root Access</i>
Basic Flow	<i>1. Open the system with the Root Access 2. Click Add Admin 3. Specify its data 4. Click submit</i>
Post-condition	<i>Admin is added</i>

4.4 Remove Admin

Use Case Name	<i>Remove Admin</i>
Actors	<i>Root — Exclusively</i>
Pre-condition	<i>Root Access</i>
Basic Flow	<ol style="list-style-type: none">1. <i>Open the system with the Root Access</i>2. <i>Click Remove Admin</i>3. <i>Choose the Admin [drop down list</i>4. <i>Click submit</i>
Post-condition	<i>Admin is removed</i>

4.5 Block Admin

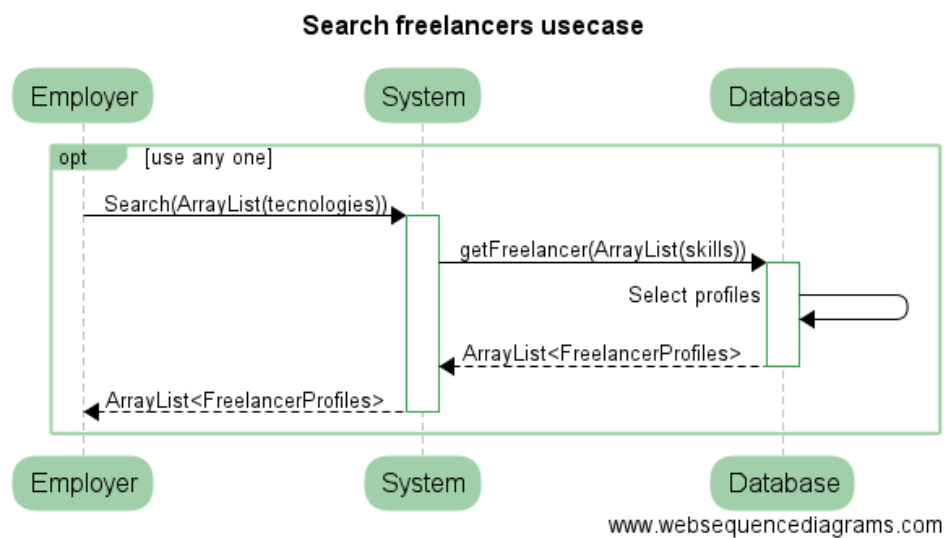
Use Case Name	<i>Block Admin</i>
Actors	<i>Root — Exclusively</i>
Pre-condition	<i>Root Access</i>
Basic Flow	<ol style="list-style-type: none">1. <i>Open the system with the Root Access</i>2. <i>Click Block Admin</i>3. <i>Choose the Admin [drop down list</i>4. <i>Click submit</i>
Post-condition	<i>Addmin is removed and is put in the block list</i>

4.6 Unblock Admin

Use Case Name	<i>Unblock Admin</i>
Actors	<i>Root — Exclusively</i>
Pre-condition	<i>Root Access</i>
Basic Flow	<ol style="list-style-type: none">1. <i>Open the system with the Root Access</i>2. <i>Click Unblock Admin</i>3. <i>Choose the Admin [drop down list</i>4. <i>Click submit</i>
Post-condition	

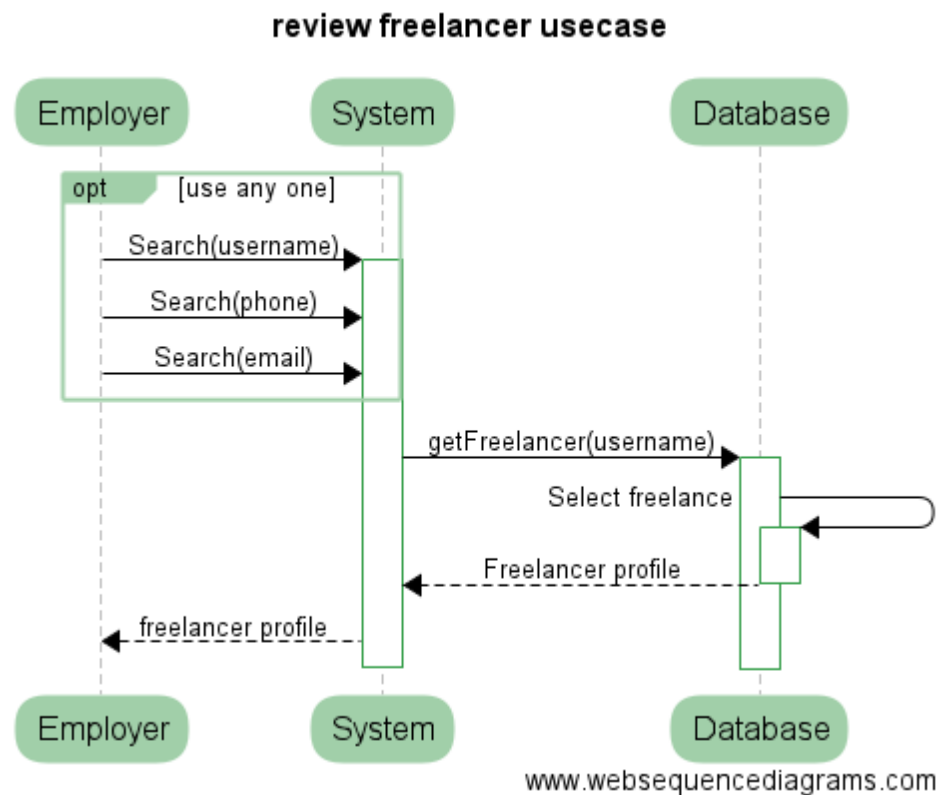
4.7 Search Freelancers

Use Case Name	<i>Search Freelancers</i>
Actors	<i>Employer — Exclusively</i>
Pre-condition	<i>Employer Access</i>
Basic Flow	<ol style="list-style-type: none"> 1. Open the system with the Employee Access 2. Click Search About Freelancers 3. set the technologies that you want 4. Click submit
Post-condition	<i>Show All freelancers that there Skills match with your Technologies</i>



4.8 Search about Freelancer

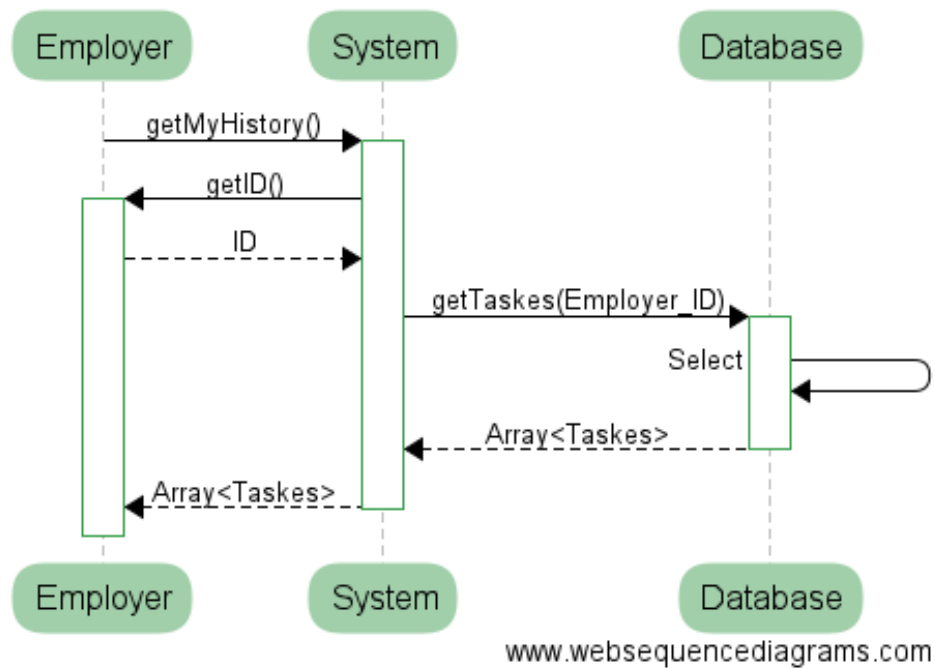
Use Case Name	<i>Search about Freelancer</i>
Actors	<i>Employer — Exclusively</i>
Pre-condition	<i>Employer Access</i>
Basic Flow	<ol style="list-style-type: none"> 1. Open the system with the Employer Access 2. Click Search About specific Freelancer 3. set his username or mail or phone 4. Click submit
Post-condition	<i>Show freelancer profile</i>



4.9 Review Employer history

Use Case Name	<i>Review Employer history</i>
Actors	<i>Employer — Exclusively</i>
Pre-condition	<i>Employer Access</i>
Basic Flow	<ol style="list-style-type: none"> 1. Open the system with the Employer Access 2. Click Show My history 3. Click submit
Post-condition	<i>Show all your Tasks and offers and his modes</i>

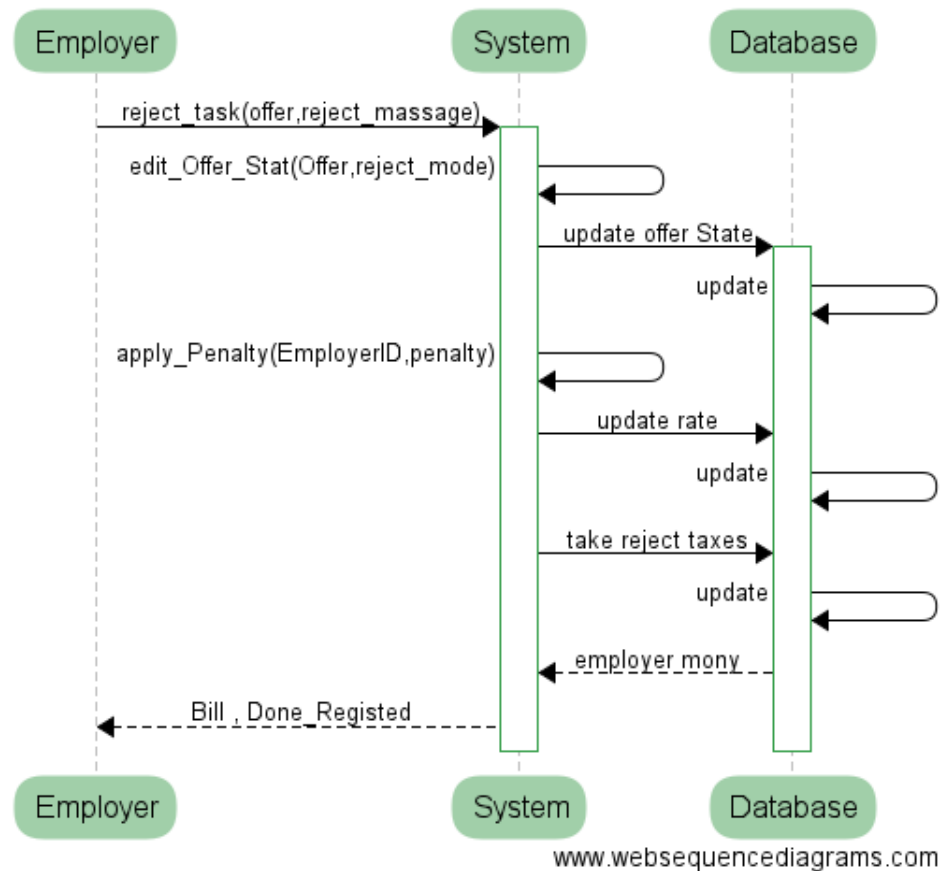
Review history usecase



4.10 Reject offer submitting

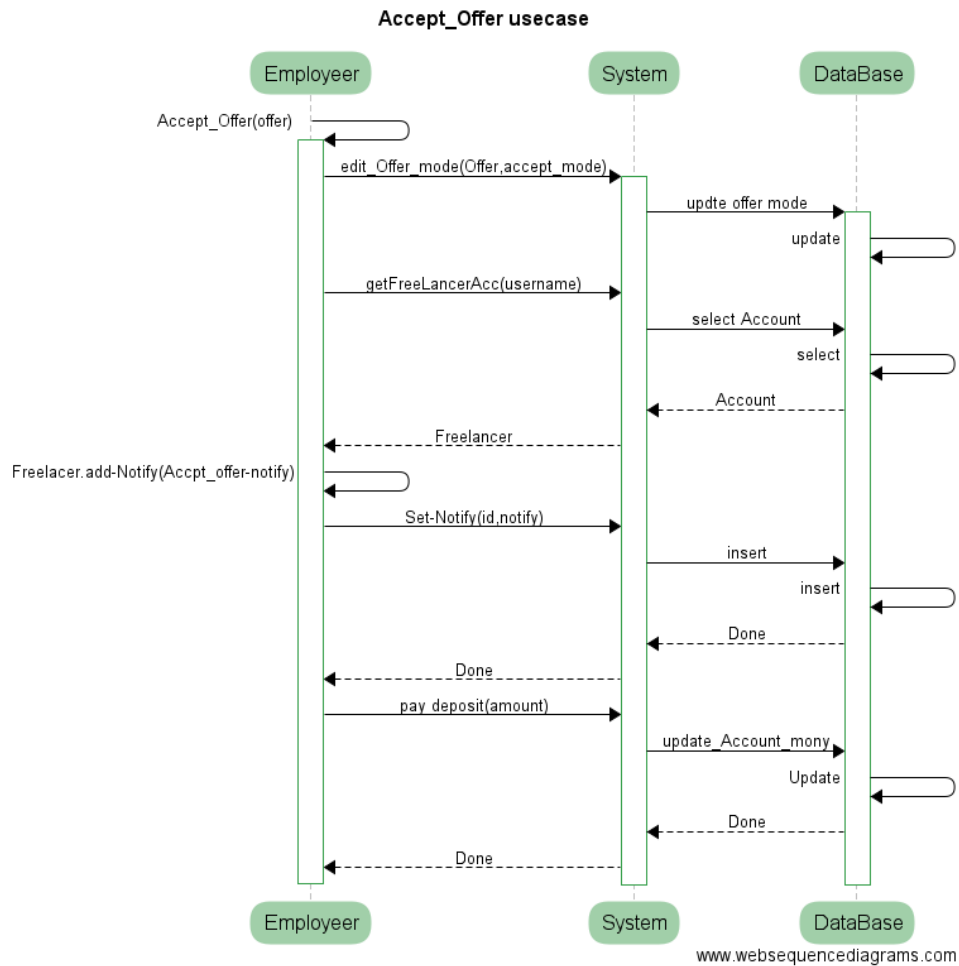
Use Case Name	<i>Reject offer submitting</i>
Actors	<i>Employer — Exclusively</i>
Pre-condition	<i>1. Employer Access</i> <i>2. Task is finished but Employer need to reject it</i>
Basic Flow	<i>1. Open the system with the Employeer Access</i> <i>2. Check submitting Offer</i> <i>3. Click reject this Offer</i>
Post-condition	<i>1. Take rejected</i> <i>2. Notify Freelancer</i> <i>3. Apply penalty</i>

Reject Offer submitting usecase



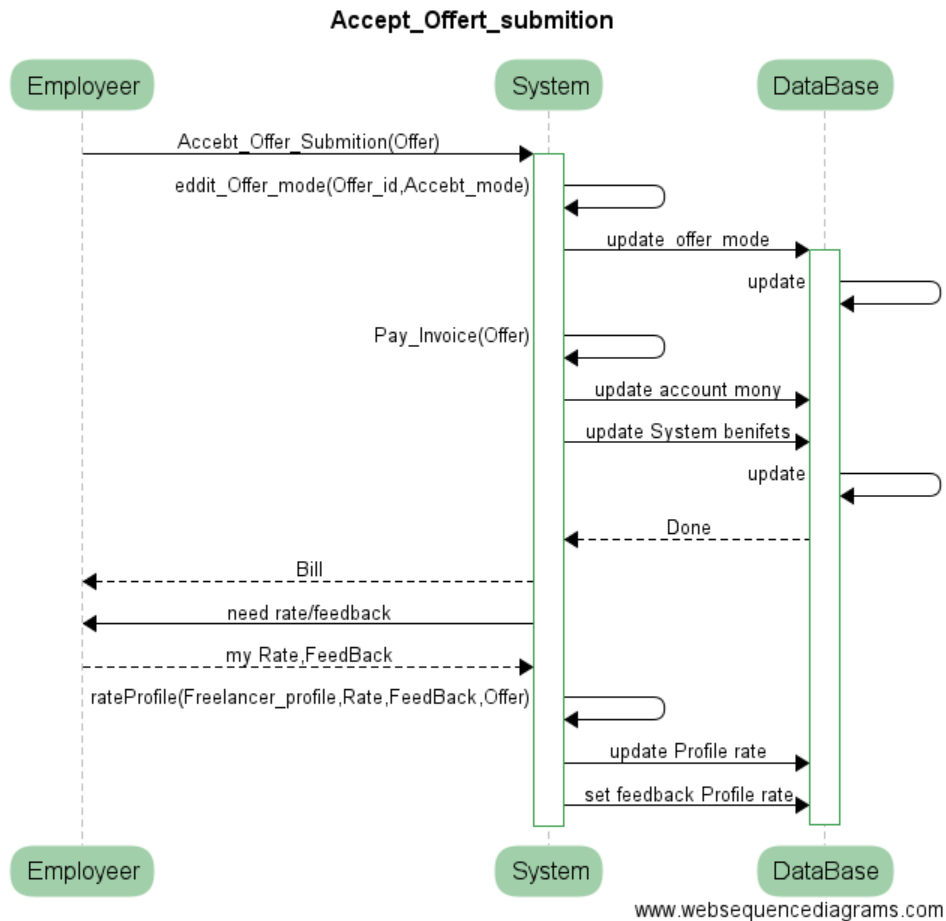
4.11 Accept Offer

Use Case Name	<i>Accept Offer</i>
Actors	<i>Employer — Exclusively</i>
Pre-condition	<i>1. Employer Access</i> <i>2. Task that have offers to do it</i>
Basic Flow	<i>1. Open the system with the Employer Access</i> <i>2. Check Take Offer</i> <i>3. select and Click show offer</i> <i>4. Click Accept Offer</i>
Post-condition	<i>1. change Tase mode</i> <i>2. Notify Freelancer to start</i> <i>3. pay deposit</i>



4.12 Accept submitting Offer

Use Case Name	<i>Accept submitting Offer</i>
Actors	<i>Employer — Exclusively</i>
Pre-condition	<i>1. Employer Access</i> <i>2. Task is finish and freelancer upload his work and task wait to submit from Employer</i>
Basic Flow	<i>1. Open the system with the Employer Access</i> <i>2. select Offer that wait submit</i> <i>3. Click submit offer and finish task</i> <i>4. Pay invoice</i> <i>5. Rate and FeedBack Freelancer profile</i>
Post-condition	<i>1. change account balance</i> <i>2. re-Count to Freelancer profile</i> <i>3. receive work</i>



5 Design Patterns

5.1 Singleton Design Pattern

Context	
Problem	
Solution	
Post-condition	

6 Call Us