

Freelancing Management System
FMS

The Team — Yaser, Umar, Taha and Boda

March 25, 2017

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
3.1	ERD	4
3.2	Use Case	5
3.3	Class Diagram	6
3.4	System Sequence Diagrams	7
4	Use Case – <i>in detail</i>	8
4.1	Access Root	8
4.2	Specify Penalties	9
4.3	Add Admin	9
4.4	Remove Admin	10
4.5	Block Admin	10
4.6	Unblock Admin	10
4.7	Search Freelancers	10
4.8	Search about Freelancer	11
4.9	Review Employer history	12
4.10	Reject offer submitting	13
4.11	Accept Offer	14
4.12	Accept Submitting Offer	15
4.13	Reply Complaints	16
4.14	View Statistics	17
4.15	Remove Account	17
4.16	Receive Complaint	17
4.17	View Reports	17
4.18	Ban Account	18
5	Design Patterns	19
5.1	Singleton Design Pattern	19
5.2	ReadOnly Pattern	20
5.3	Factory Pattern	21
5.4	Composition Pattern	22
5.5	Bridge Design Pattern	23
6	Call Us	23

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*.

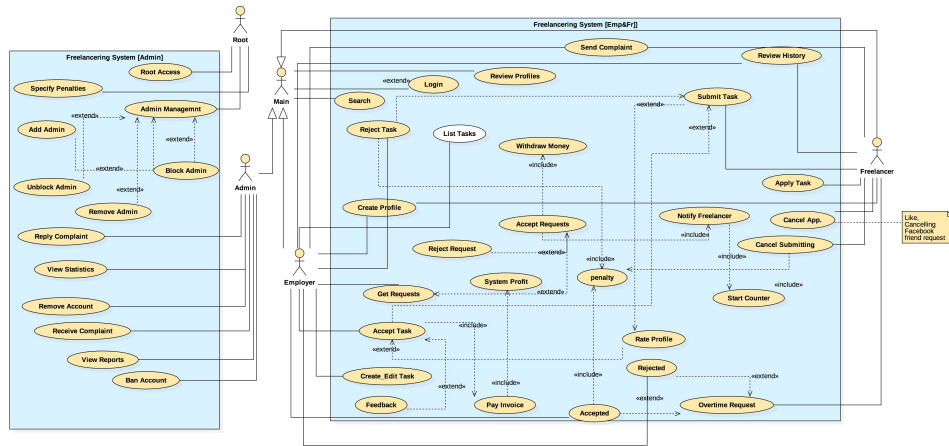
2.4 Employer

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.

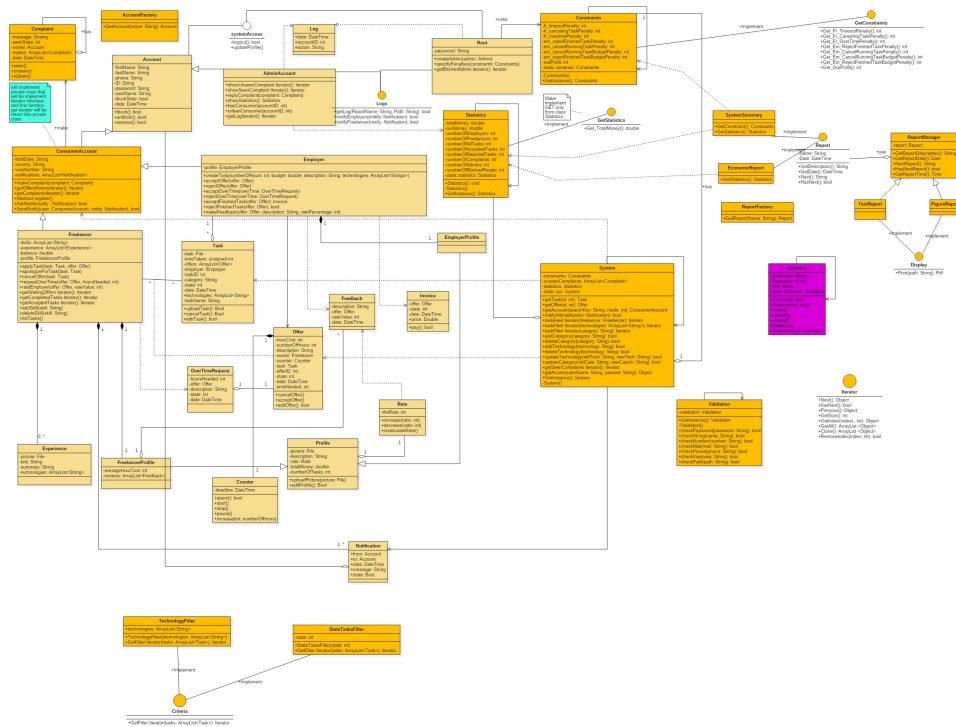
3.1 ERD



3.2 Use Case



3.3 Class Diagram



3.4 System Sequence Diagrams

Figure 1: "System Sequence of the Root"

Figure 2: System Sequence Diagram of the *Root*

Figure 3: System Sequence Diagram of the *Employer*

Figure 4: System Sequence Diagram of the *Freelancer*

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 configured</i>

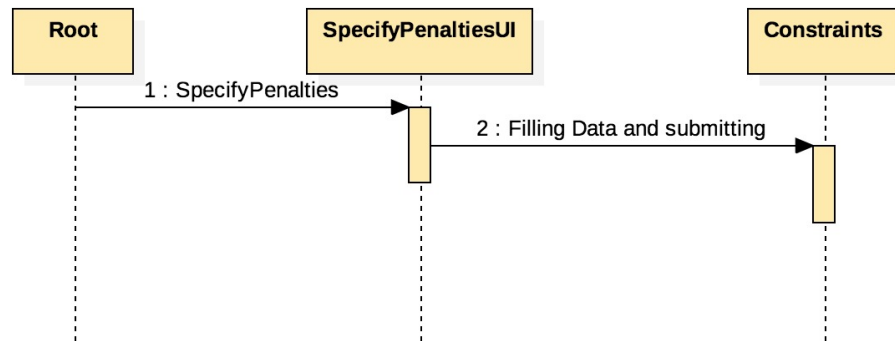


Figure 5: Sequence Diagram of the *Specifying Penalties*

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. Open the system with the Root Access2. Click Remove Admin3. Choose the Admin [drop down list4. Click submit
Post-condition	<i>Admin is removed</i>

4.5 Block Admin

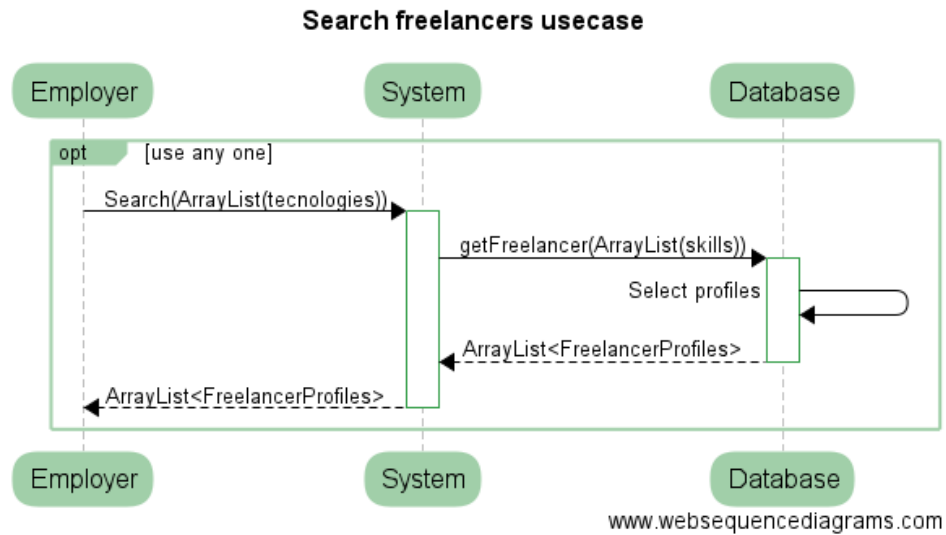
Use Case Name	<i>Block Admin</i>
Actors	<i>Root — Exclusively</i>
Pre-condition	<i>Root Access</i>
Basic Flow	<ol style="list-style-type: none">1. Open the system with the Root Access2. Click Block Admin3. Choose the Admin [drop down list4. Click submit
Post-condition	<i>Addmin is removed and is put 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. Open the system with the Root Access2. Click Unblock Admin3. Choose the Admin [drop down list4. Click submit
Post-condition	

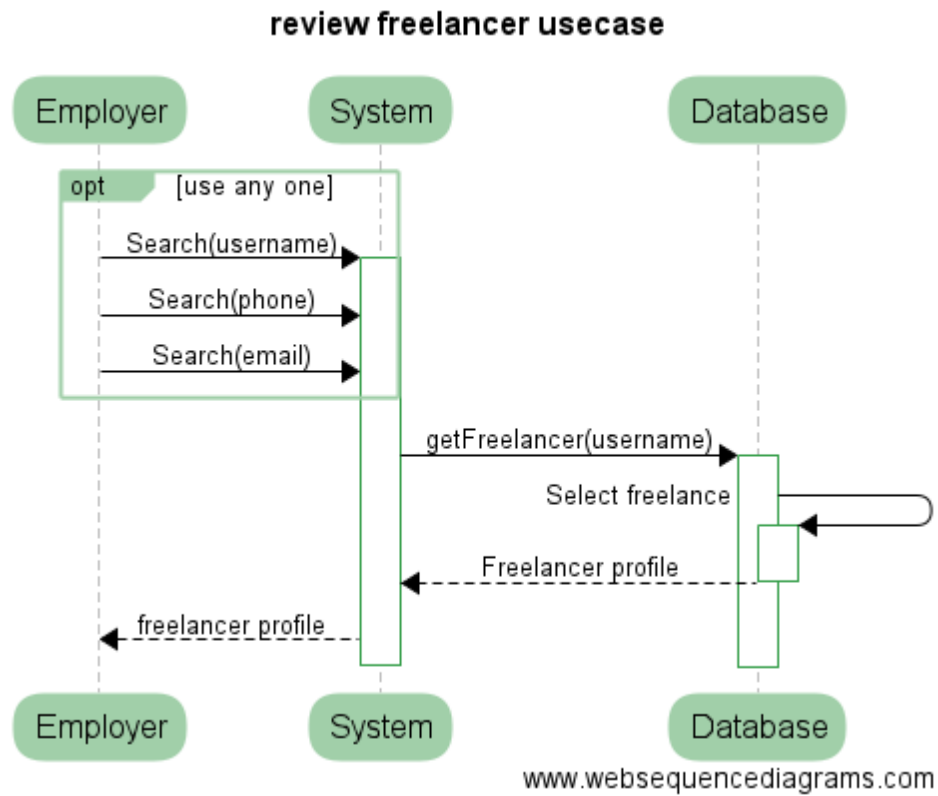
4.7 Search Freelancers

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 Employeer Access2. Click Search About Freelancers3. set the technologies that you want4. Click submit
Post-condition	<i>Show All freelancers that there Skills mathch with your Technologies</i>



4.8 Search about Freelancer

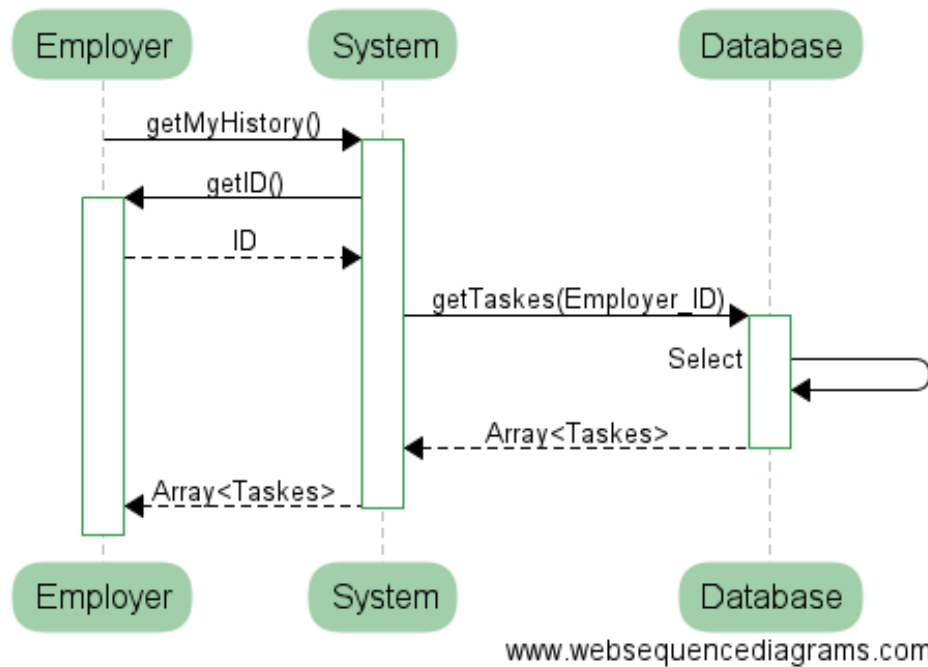
Use Case Name	<i>Search about Freelancer</i>
Actors	<i>Employer — Exclusively</i>
Pre-condition	<i>Employer Access</i>
Basic Flow	<ol style="list-style-type: none"> 1. Open the system with the Employer Access 2. Click Search About specific Freelancer 3. set his username or mail or phone 4. Click submit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



4.9 Review Employer history

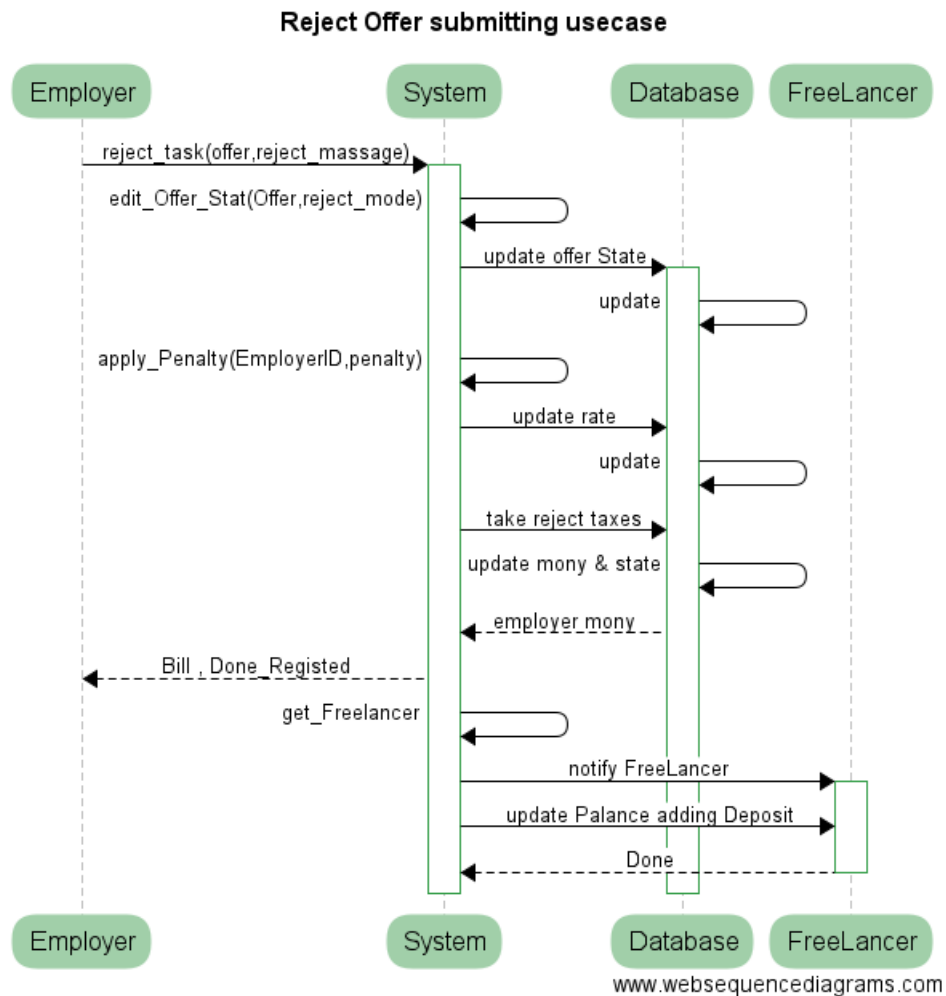
Use Case Name	<i>Review Employer history</i>
Actors	<i>Employer — Exclusively</i>
Pre-condition	<i>Employer Access</i>
Basic Flow	<ol style="list-style-type: none"> 1. <i>Open the system with the Employer Access</i> 2. <i>Click Show My history</i> 3. <i>Click submit</i>
Post-condition	<i>Show all your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

Review history usecase



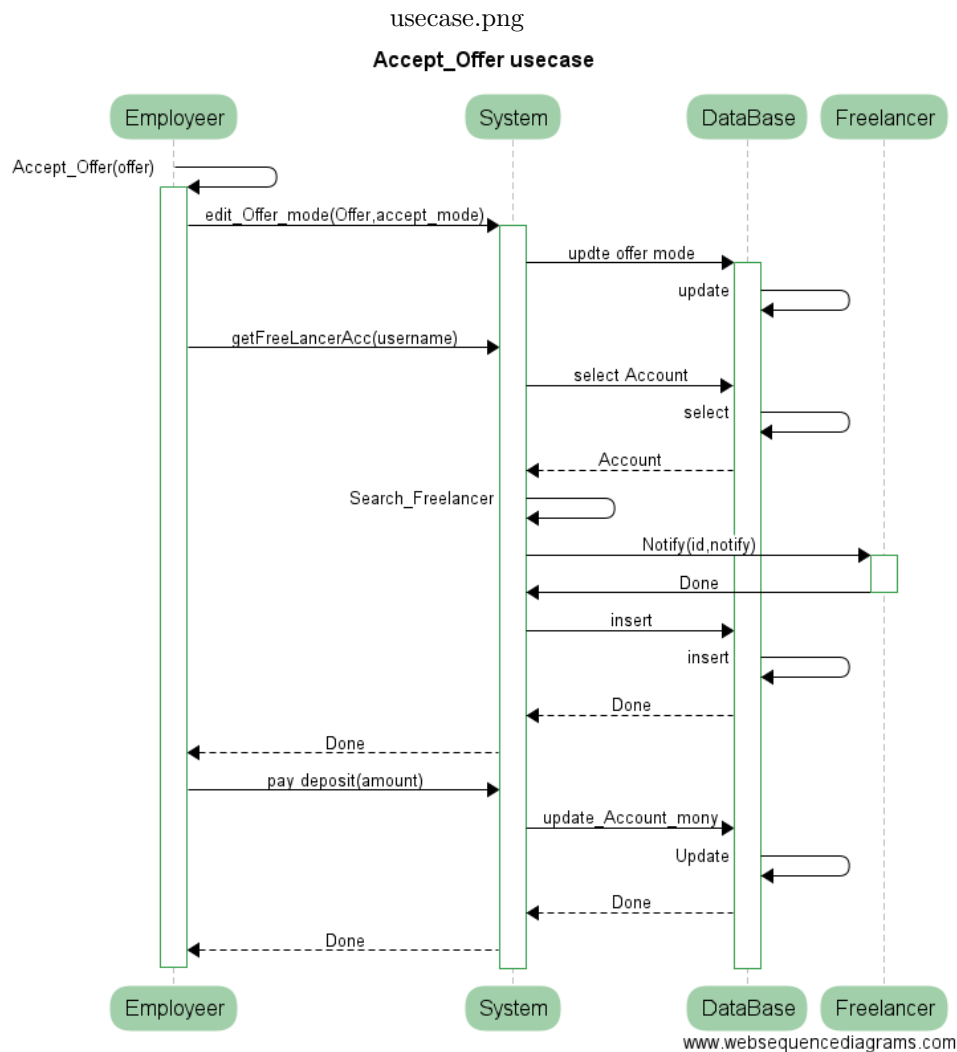
4.10 Reject offer submitting

Use Case Name	<i>Reject offer submitting</i>
Actors	<i>Employer — Exclusively</i>
Pre-condition	<ol style="list-style-type: none"> 1. <i>Employer Access</i>
Basic Flow	<ol style="list-style-type: none"> 1. <i>Open the system with the Employeer Access</i> 2. <i>Check submitting Offer</i> 3. <i>Click reject this Offer</i>
Post-condition	<ol style="list-style-type: none"> 1. <i>Take rejected</i> 2. <i>Notify Freelancer</i> 3. <i>Apply penalty</i>



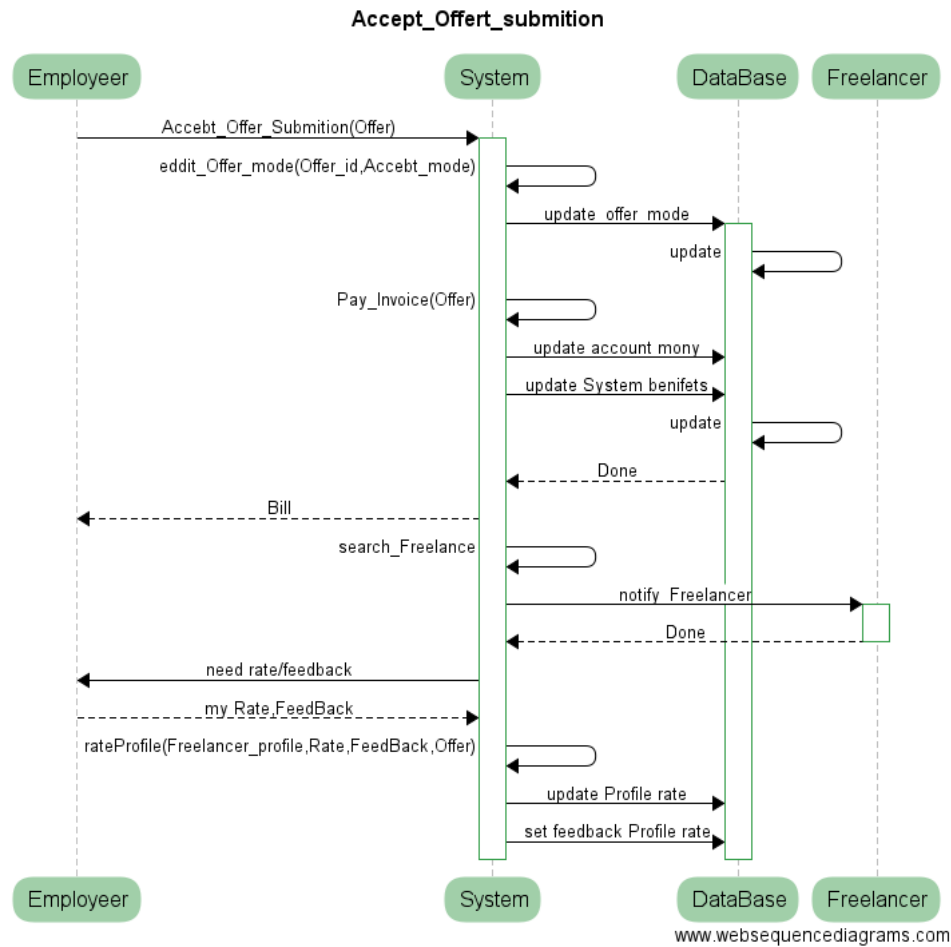
4.11 Accept Offer

Use Case Name	<i>Accept Offer</i>
Actors	<i>Employer — Exclusively</i>
Pre-condition	<ol style="list-style-type: none"> 1. <i>Employer Access</i> 2. <i>Task that have offers to do it</i>
Basic Flow	<ol style="list-style-type: none"> 1. <i>Open the system with the Employer Access</i> 2. <i>Check Take Offer</i> 3. <i>select and Click show offer</i> 4. <i>Click Accept Offer</i>
Post-condition	<ol style="list-style-type: none"> 1. <i>change Tase mode</i> 2. <i>Notify Freelancer to start</i> 3. <i>pay deposit</i>



4.12 Accept Submitting Offer

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



4.13 Reply Complaints

Use Case Name	<i>Reply Complaints</i>
Actors	<i>Admin</i>
Pre-condition	<ol style="list-style-type: none"> 1. <i>Admin Access</i> 2. <i>Freelancer/Employer has sent a Complaint</i>
Basic Flow	<ol style="list-style-type: none"> 1. <i>Open the system with the Admin Access</i> 2. <i>Select Show All Complaints from the sidebar</i> 3. <i>Select the complaint and click to reply</i> 4. <i>Write your reply and submit</i>
Post-condition	<i>A notification has sent to the sender of the commitment</i>

4.14 View Statistics

Use Case Name	<i>View Statistics</i>
Actors	<i>Admin</i>
Pre-condition	1. <i>Admin Access</i> 2. <i>Freelancer/Employer has sent a Complaint</i>
Basic Flow	1. <i>Open the system with the Admin Access</i> 2. <i>Select View Statistics from the sidebar</i> 3. <i>Now, You can select one of the statistics perspective</i>
Post-condition	<i>A graph is appeared</i>

4.15 Remove Account

Use Case Name	<i>Remove Account</i>
Actors	<i>Admin</i>
Pre-condition	1. <i>Admin Access</i> 2. <i>Freelancer/Employer has sent a Complaint</i>
Basic Flow	1. <i>Open the system with the Admin Access</i> 2. <i>Select Show All Freelancer/Employer from the sidebar</i> 3. <i>Select the one you want to removed</i> 4. <i>Select Remove Account</i> 5. <i>Confirm!</i>
Post-condition	<i>An Account it removed</i>

4.16 Receive Complaint

Use Case Name	<i>Receive Complaint</i>
Actors	<i>Admin</i>
Pre-condition	1. <i>Admin Access</i> 2. <i>Freelancer/Employer has sent a Complaint</i>
Basic Flow	1. <i>Open the system with the Admin Access</i> 2. <i>You will find a new notification</i>
Post-condition	<i>You've been notified</i>

4.17 View Reports

Use Case Name	<i>View Reports</i>
Actors	<i>Admin</i>
Pre-condition	1. <i>Admin Access</i> 2. <i>Freelancer/Employer has sent a Complaint</i>
Basic Flow	1. <i>Open the system with the Admin Access</i> 2. <i>Select View Reports</i> 3. <i>Select the Subject of the Report</i>
Post-condition	<i>You have an up-to-date report, you can get the PDF version</i>

4.18 Ban Account

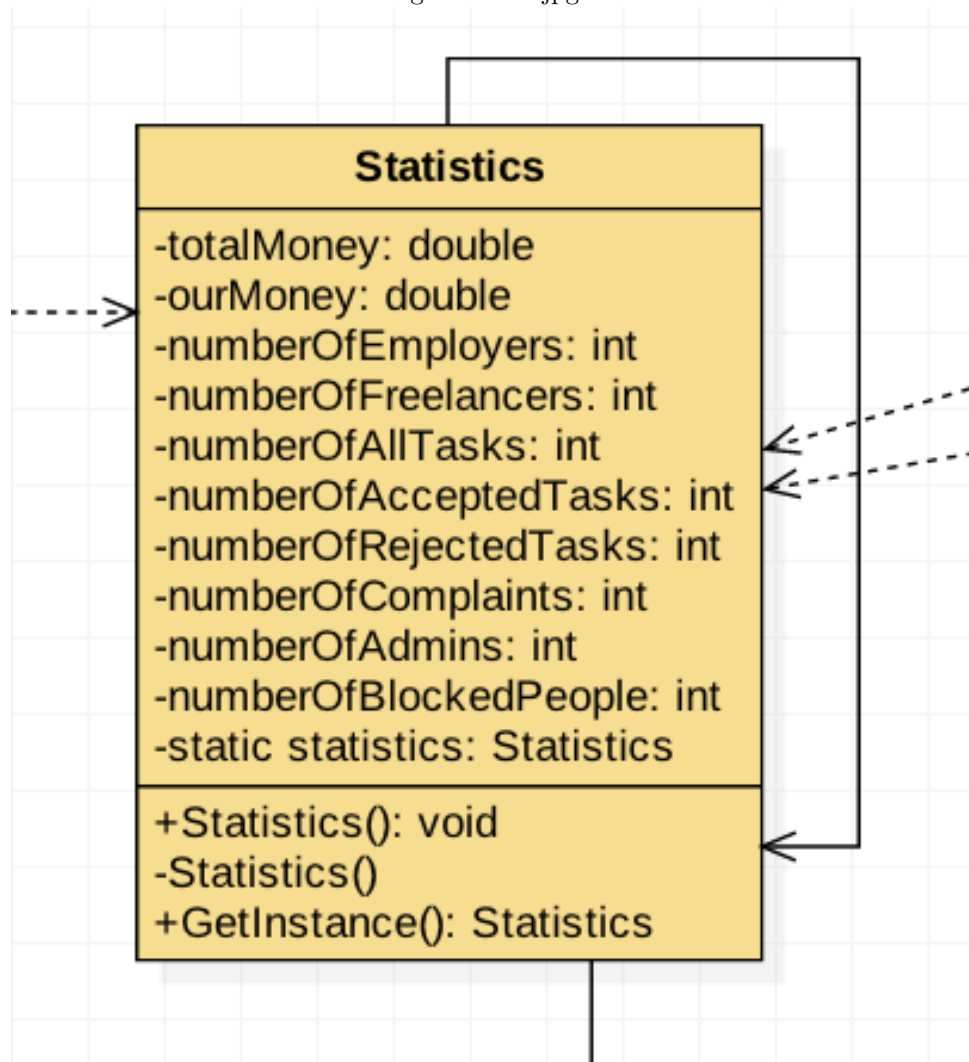
Use Case Name	<i>Ban Account</i>
Actors	<i>Admin</i>
Pre-condition	<i>1. Admin Access 2. Freelancer/Employer has sent a Complaint</i>
Basic Flow	<i>1. Open the system with the Admin Access 2. Select Show All Freelancer/Employer from the sidebar 3. Select the one you want to Ban 4. Select Ban Account 4. Confirm!</i>
Post-condition	<i>An Account is Banned</i>

5 Design Patterns

5.1 Singleton Design Pattern

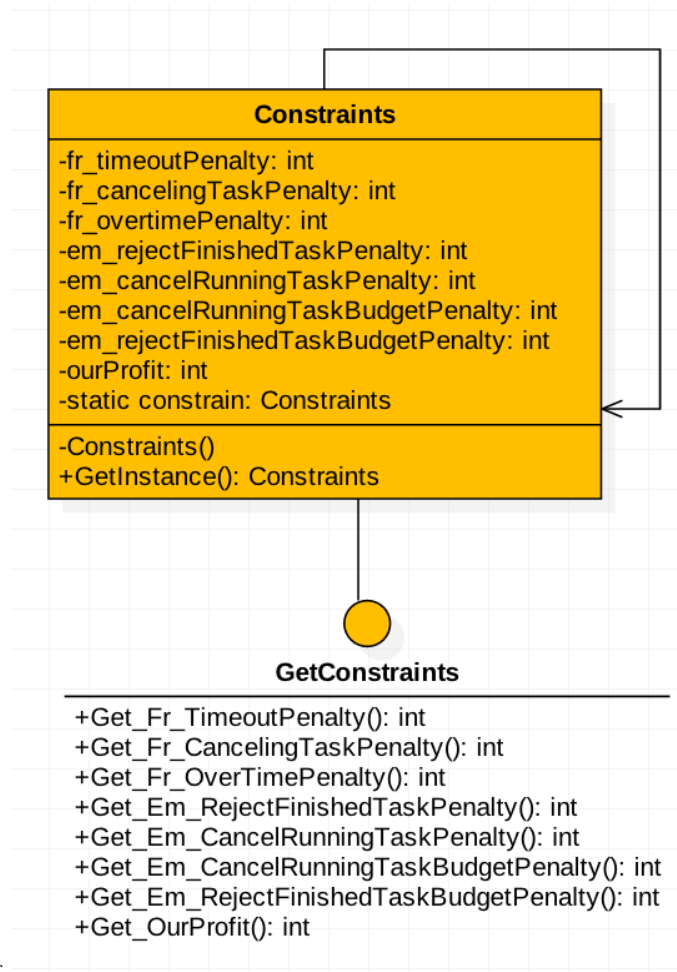
Context	We need to get only one instance (Statistics/Database Class...)
Problem	How can we insure that it is not possible to get more than one instance of a Class
Solution	1. <i>Make a the constructor private.</i> 2. <i>Make public static method, It creates single object.</i>

Design Pattern.jpg



5.2 ReadOnly Pattern

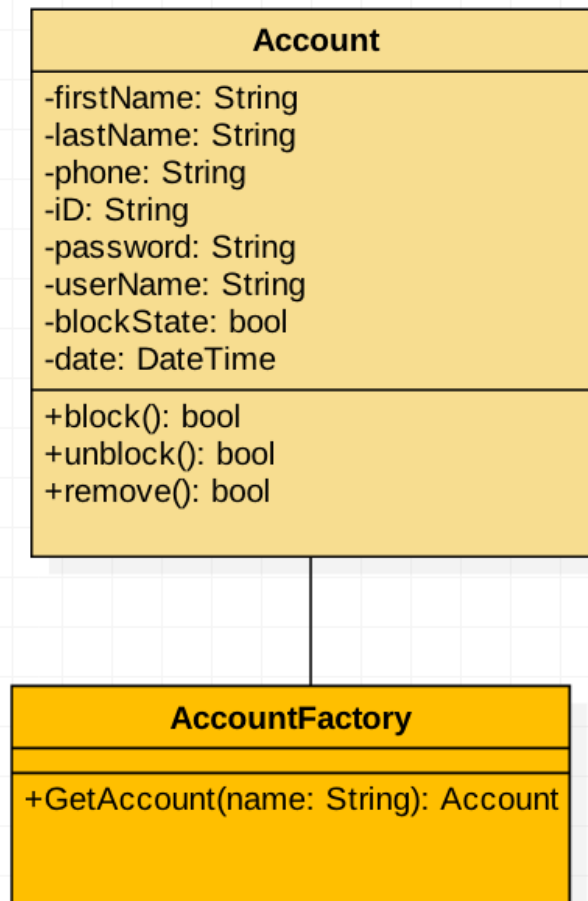
Context	We need an object to be read-only (We cannot edit it) [Statistics].
Problem	How do you create a class whose instances are immutable.
Solution	1. Make an Interface 'Read-Only' with getters methods.



Design Pattern.png

5.3 Factory Pattern

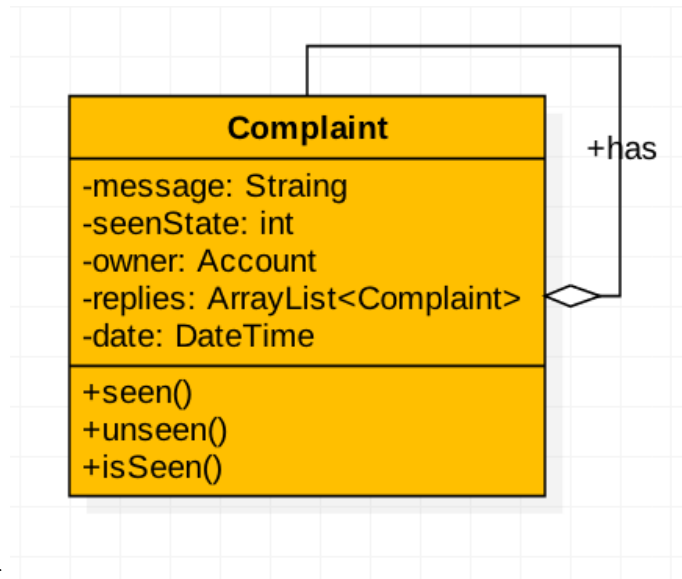
Context	We need to generate objects as a template (automatically).
Problem	How to generate objects using a generating method (Factory).
Solution	<i>Make a class (Factory), that contains a method which returns, an object of the wanted class</i>



Design Pattern.png

5.4 Composition Pattern

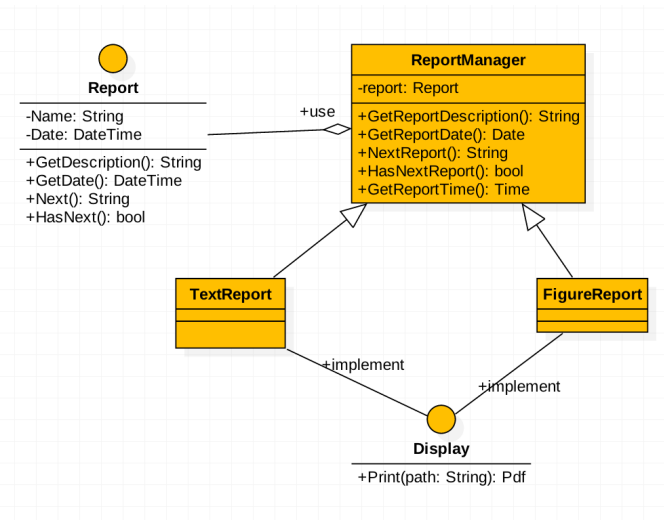
Context	Recursive composition, Something that can contain it self.
Problem	How can we Enable the complaint to contain a reply (<i>Which is a complain</i>).
Solution	<i>It can be done by self-composition</i>



Design Pattern.png

5.5 Bridge Design Pattern

Context	We need abstraction, and it can be implemented in different ways.
Problem	We have <i>Report class</i> , it has an abstraction, but there are different subjects of a Report.
Solution	<i>Make a Report Class, that implements an interface (abstraction)</i> <i>All different types of the report are children of the Report</i> <i>Each of them implement the abstraction in its own way</i>



Design Pattern.png

6 Call Us

THE END