Flextask

Ramzi Haddad 100885168

Michael Jimma 100963147

Contents

Flextask summary 3
System sequence diagrams 4
Three Layered Sequence Diagrams 7
Design class diagrams 10
Method contracts 11
Method Specifications 12
ER diagram 15
Data Access Speed Optimization strategies 16
Use scenarios 17
Windows Navigation Diagram (WND) 18
Interface Design Prototype 19
Deployment Diagram 22

Flextask summary

Flextask is a virtual platform that allows people to find person to person. People can post task online and hire some other to the job for them. Flextask is unique platform because it allows people to share their jobs freely and securely. Flextask is special because it's the only platform that allow exchange jobs and post freely without using any credit cards or any disturbing payment method. In fact, people would prefer Flextask to share their jobs because they don't have to worry about company charge, taxes or any other problems. people can do each other jobs for a specific amount of money after being in reverse auction to choose the lowest amount of money and the best workers.

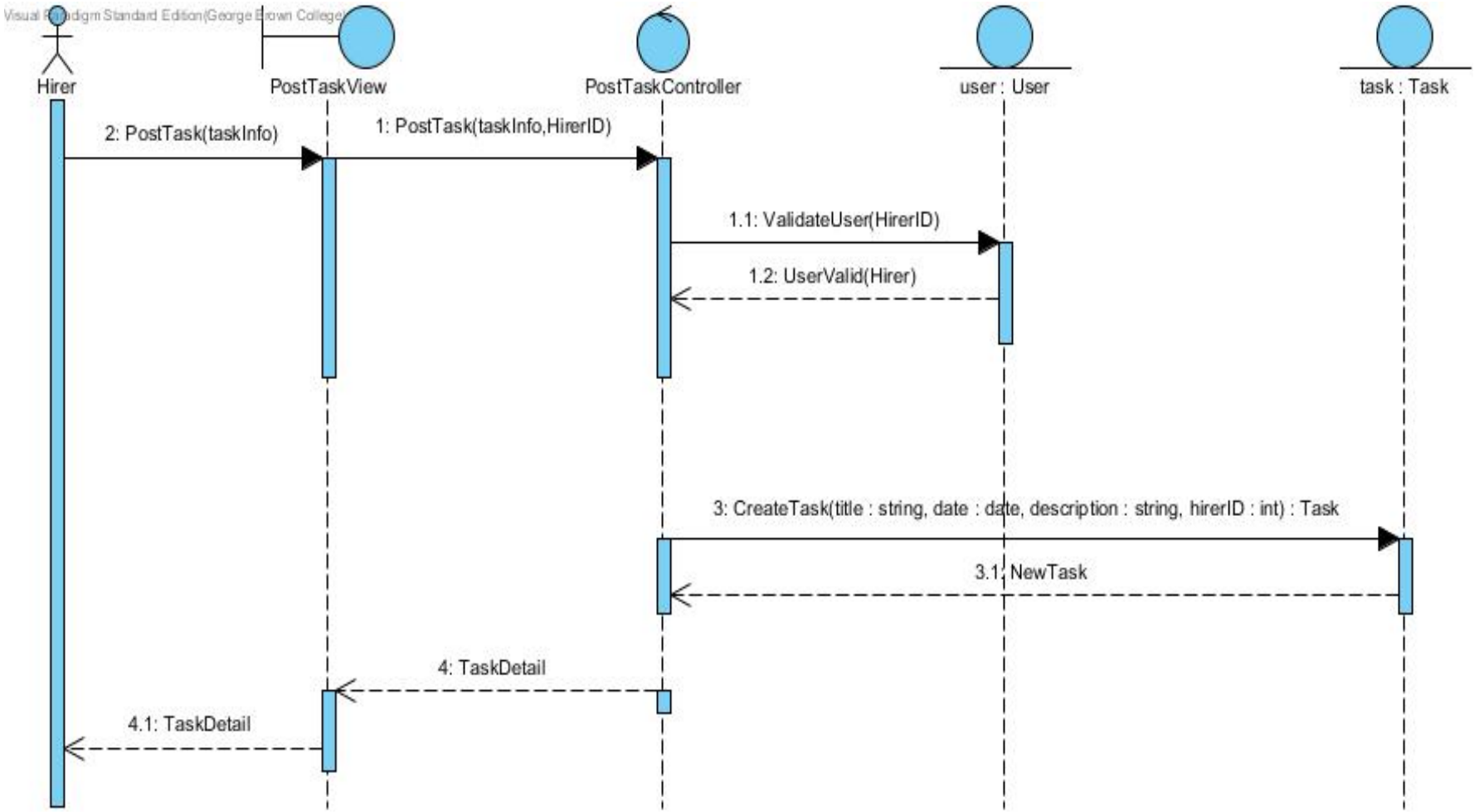
During the analyses phase we have created the project's plan. We have determined the feasibility of the project of all sides including functional, financial and operational feasibility. In addition, we have created the sequence diagrams, state-machine diagrams, scenario, and CRUID MATRIX diagrams based on the use case diagrams to analyse the behaviour of the system, methods implantation and the method and the data type for class implantation. We have created a specific design in the sequence that determine the interaction between the object and the system to achieve the target goal.

Also we have determined that if the project is possible to be made and what are the impact of it on the employees and how is it going to increase the income of the company and how long it would take to get the return of investment, also, we have calculated the impact of this project on the environment. We have recorded the behaviour of the objects and how are they going to interact together using the sequence diagrams and the machine state diagram. Also we have determined the methods and the attributes in the class diagrams. In addition, we have drawn the first shape of the behaviour as a summary of the diagrams.

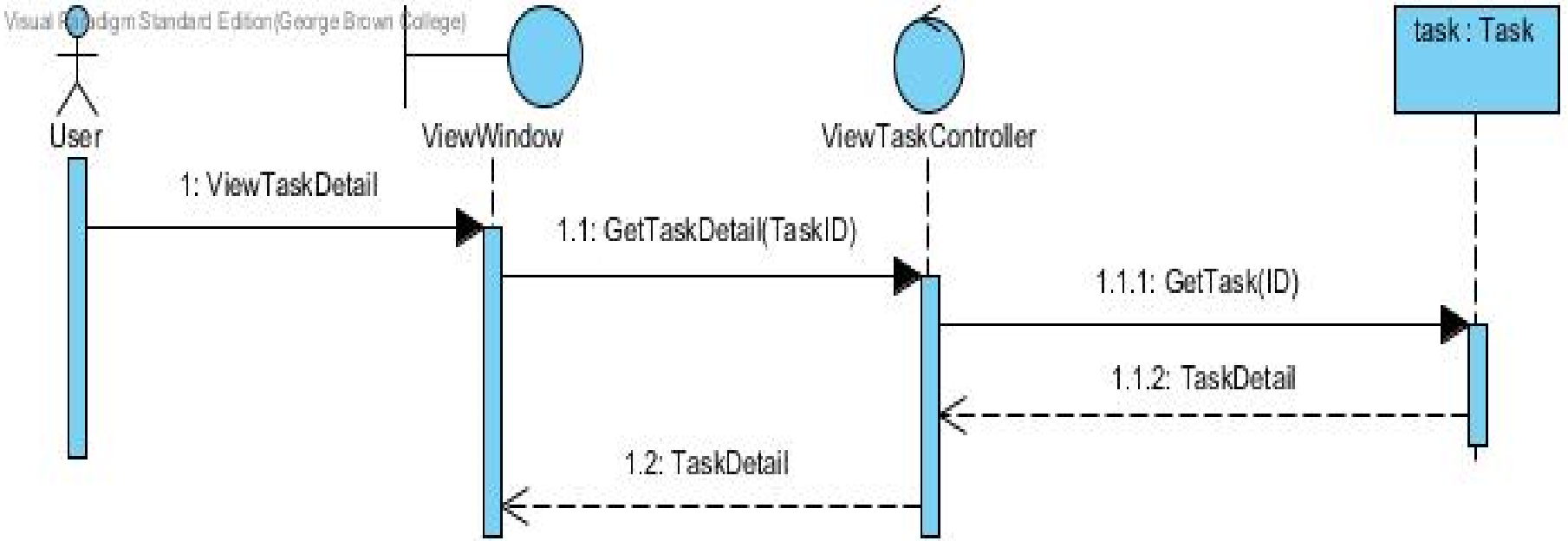
We have also created a work plan that specify the duration of the creation of the system in addition, we have assigned duties for each person who is involved in developing this project. We have calculated the time needed to finish the project and the actual cost of the project using Microsoft project.

System sequence diagrams

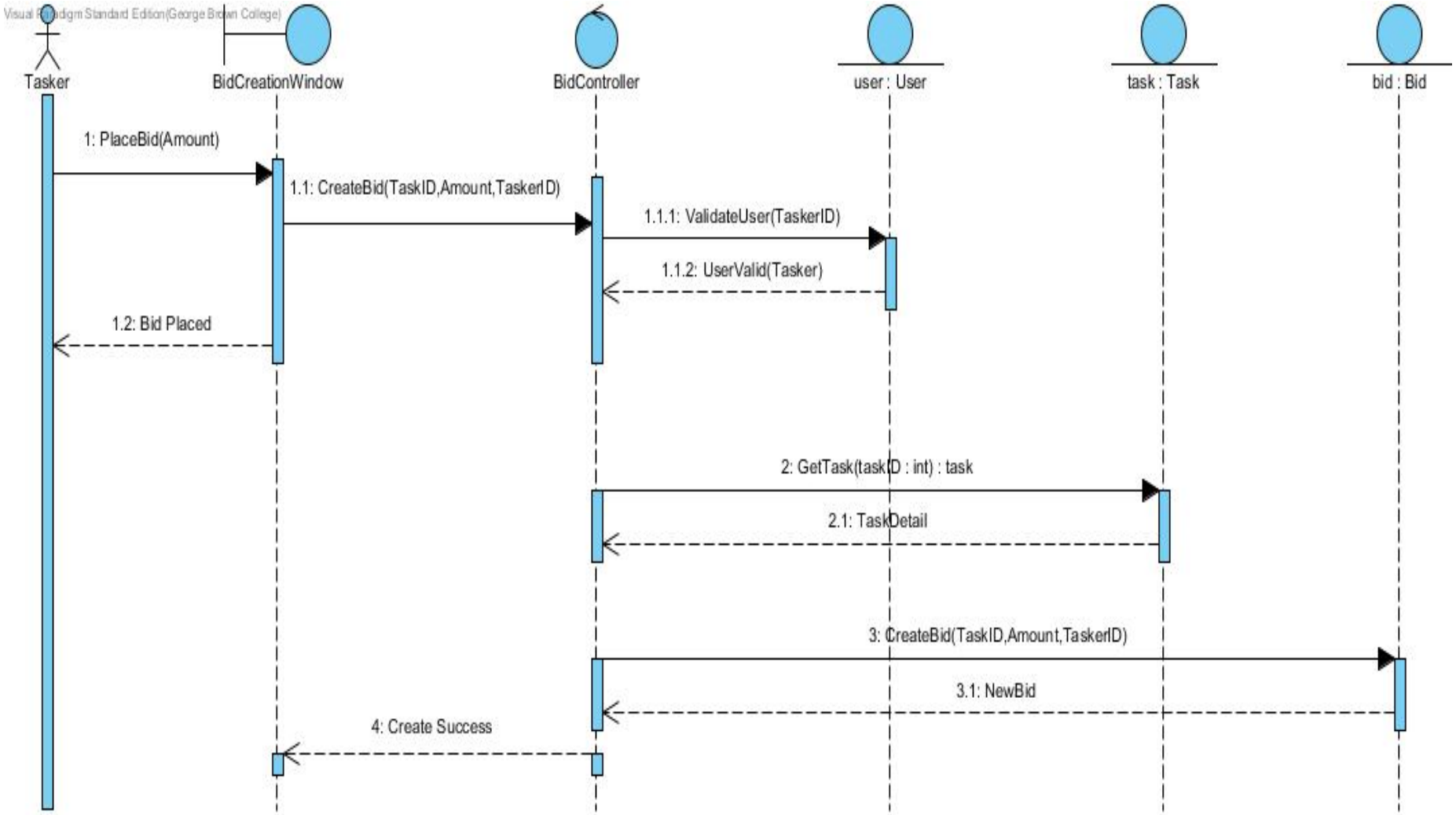
Post task usecase



ViewTask usecase

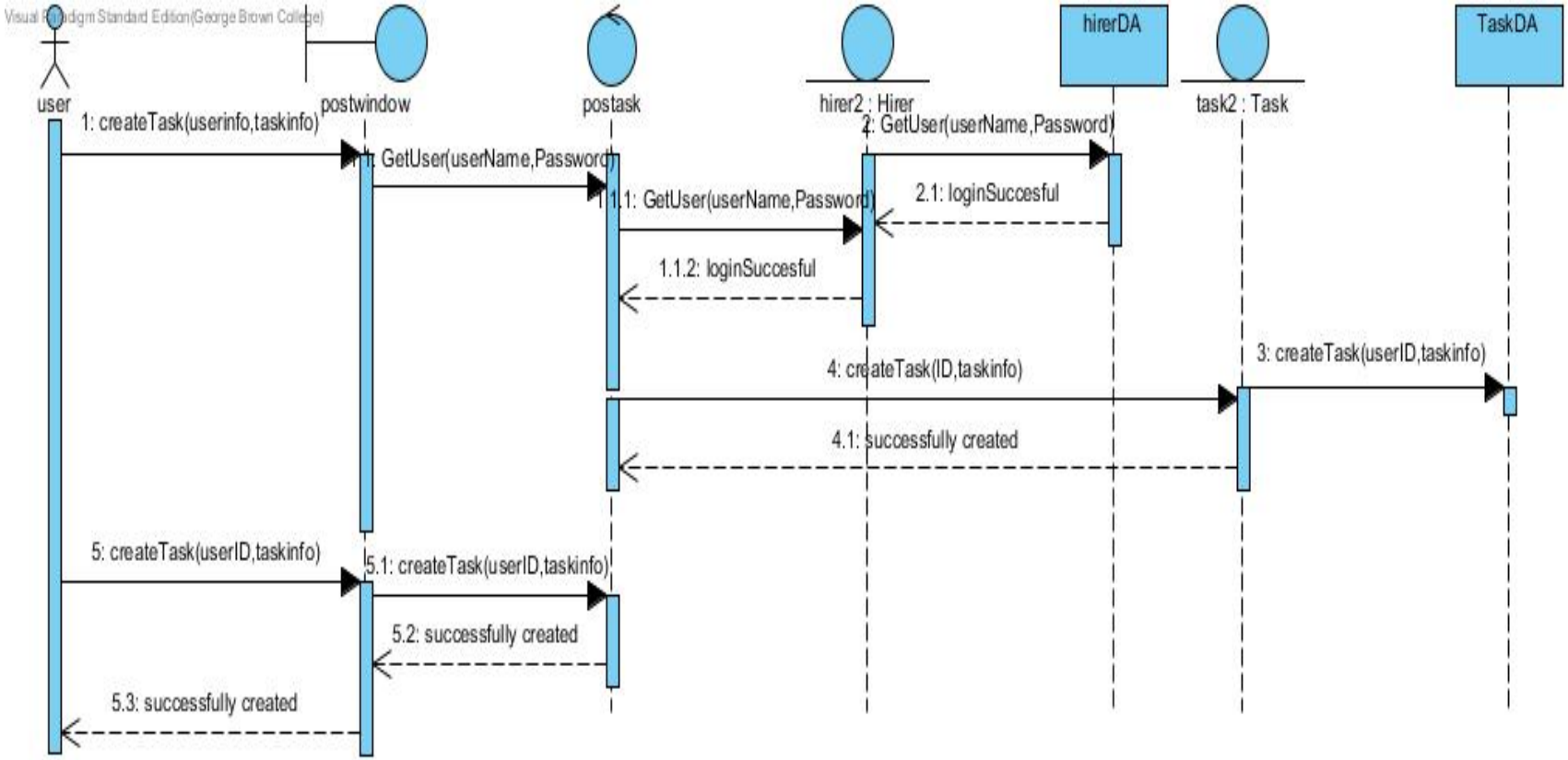


Bid usecase

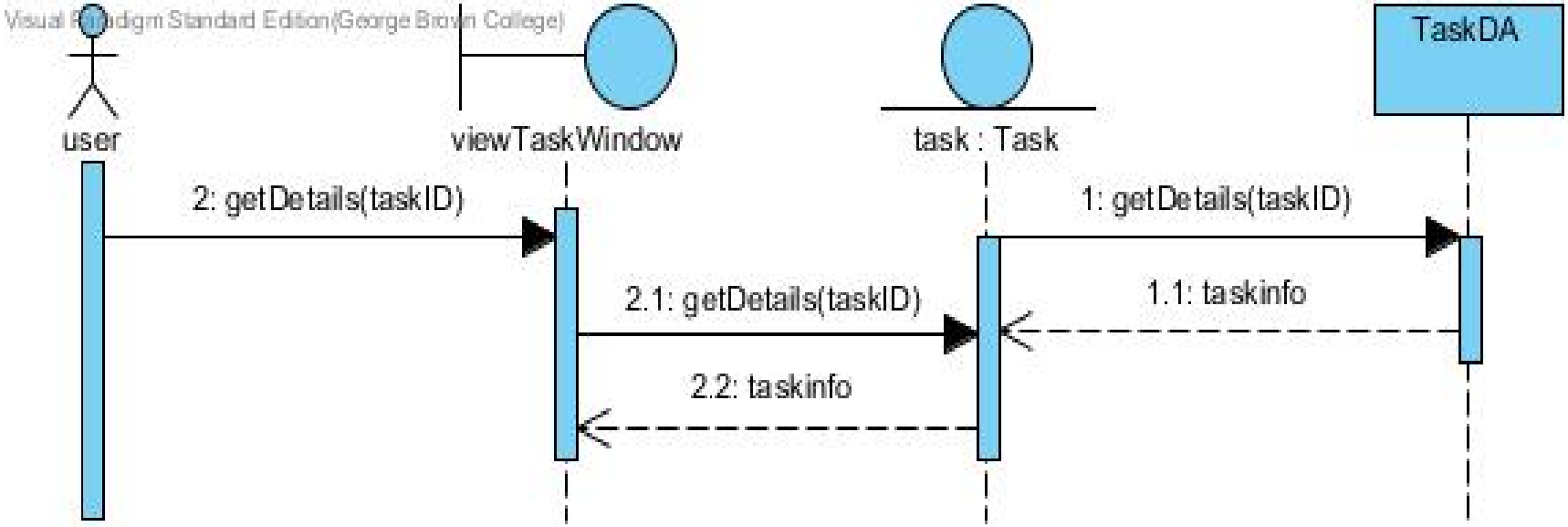

Tasker
BidCreationWindow
BidController
user : User
task : Task
bid : Bid
1: PlaceBid(Amount)
1.1: CreateBid(TaskID,Amount,TaskerID)
1.1.1: ValidateUser(TaskerID)
1.1.2: UserValid(Tasker)
1.2: Bid Placed
2: GetTask(taskID : int) : task
2.1: TaskDetail
3: CreateBid(TaskID,Amount,TaskerID)
3.1: NewBid
4: Create Success

Three Layered Sequence Diagrams

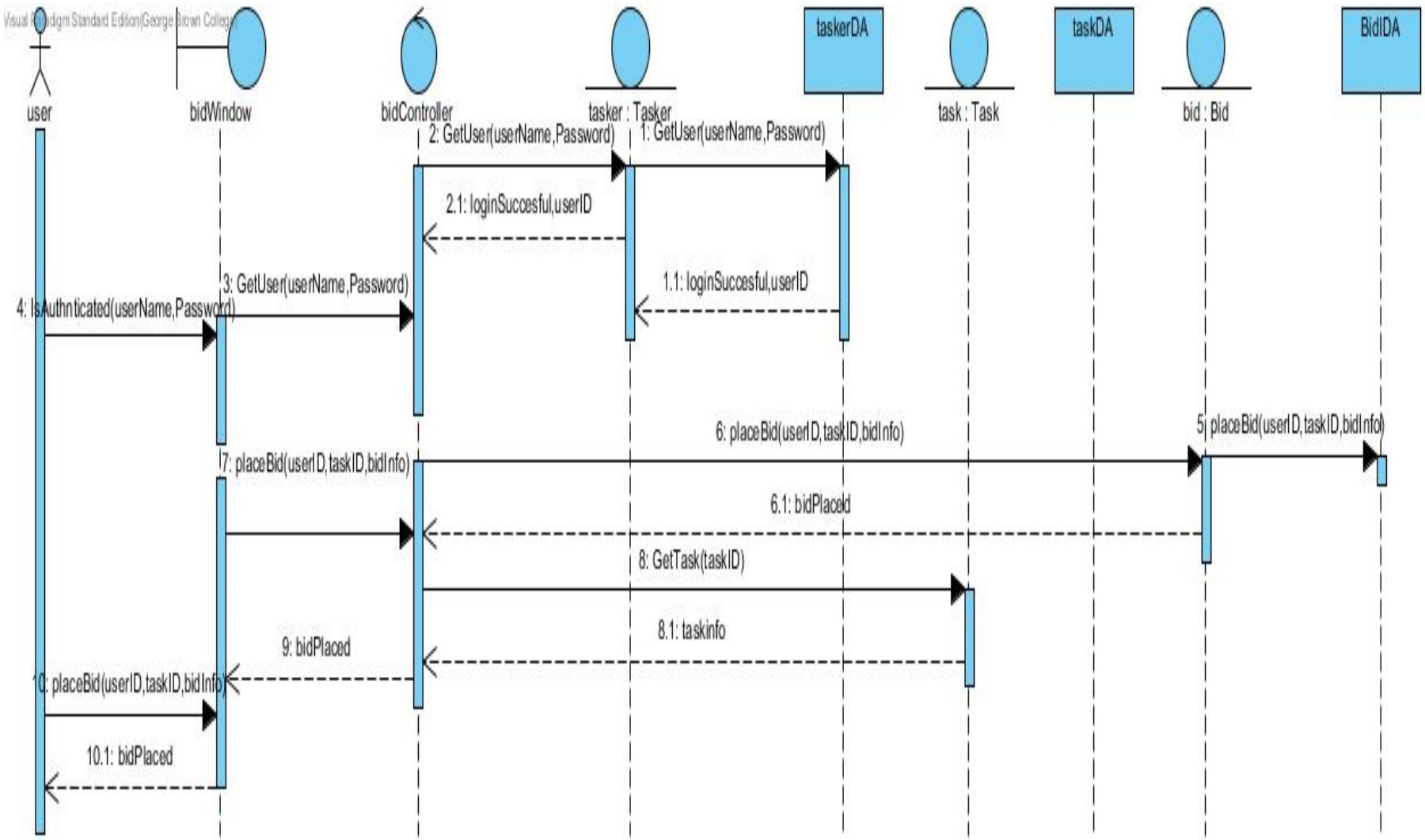
Post task usecase



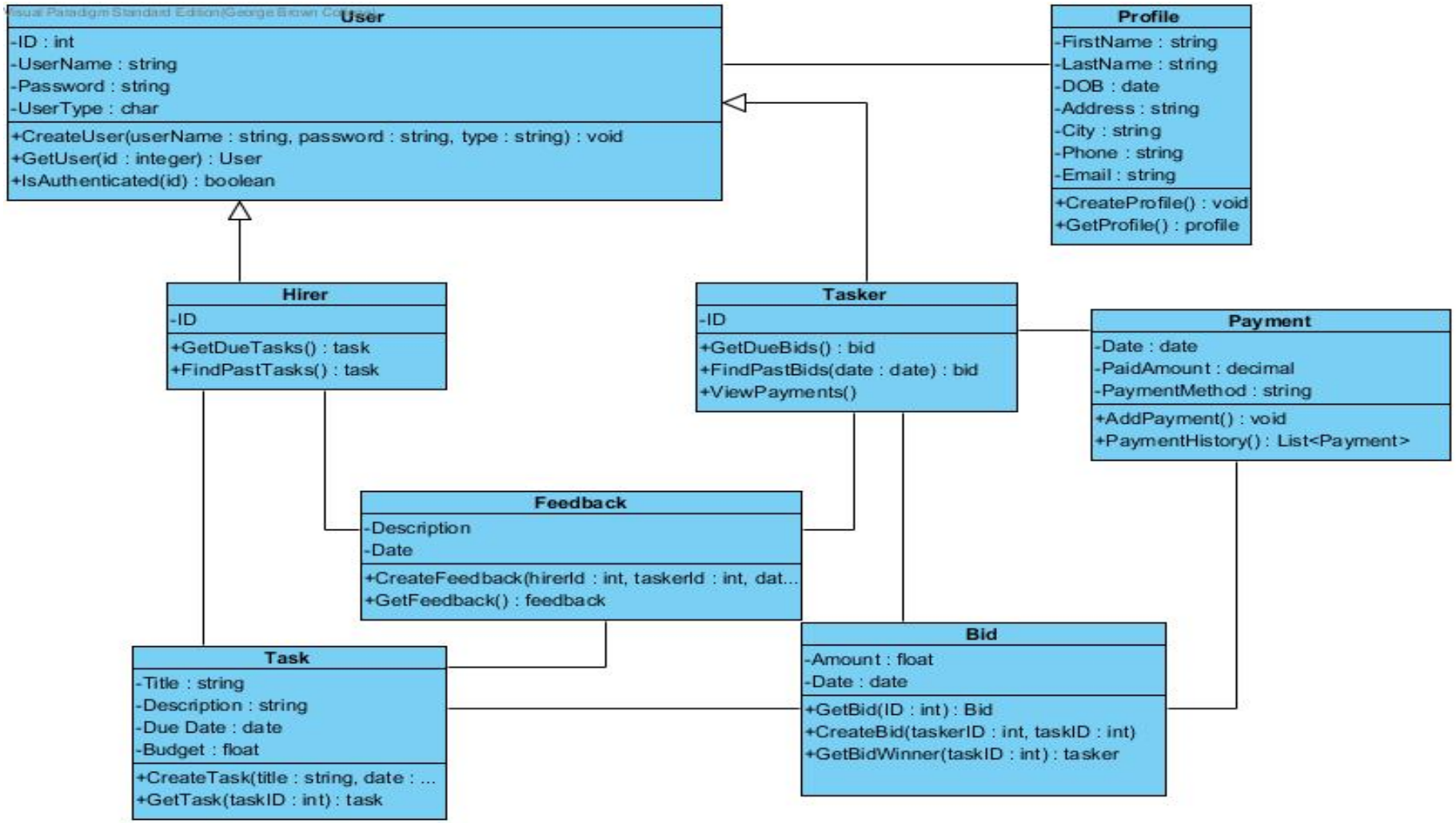
viewTask usecase



Bid usecase


user
bidWindow
bidController
tasker : Tasker
taskerDA
task : Task
taskDA
bid : Bid
BidIDA
2: GetUser(userName,Password)
1: GetUser(userName,Password)
2.1: loginSuccesful,userID
3: GetUser(userName,Password)
1.1: loginSuccesful,userID
4: IsAuthnticated(userName,Password)
6: placeBid(userID,taskID,bidInfo)
5: placeBid(userID,taskID,bidInfo)
7: placeBid(userID,taskID,bidInfo)
6.1: bidPlaced
8: GetTask(taskID)
8.1: taskinfo
9: bidPlaced
10: placeBid(userID,taskID,bidInfo)
10.1: bidPlaced

Design class diagrams



Method contracts

Method Name: GetUser	**Class Name:** User	**ID:**101
Client: task		
Associated use Case: post a task		
Description of responsibilities: this method will look for the user to verify identity and account of the hirer		
Arguments received: String username, string password		
Type of valid Returned: boolean		
Pre-condition: none		
Post-condition: none		

Method Name: createTask	**Class Name:** task	**ID:** 102
Client: task		
Associated use Case: post a task		
Description of responsibilities: this method will be responsible of creating a new post and add the task in task table		
Arguments received: userID, taskinfo		
Type of valid Returned: none		
Pre-condition: userID!=null		
Pre-condition:none		

Method Name: getDetails	**Class Name:** task	**ID:** 104
Client: task		
Associated use Case: view task		
Description of responsibilities: this method will display the details of the select task		
Arguments received: taskID		
Type of valid Returned: task:task		
Pre-condition:none		
Pre-condition:none		

Method Specifications

Method Name: GetUser	**Class Name:** User	**ID:**301
Contract ID:101	**Programmer:** Haddad,Ramzi	**Date Due: 1/05/2016**
Programming Language: ***PHP**		
Triggers/Events: the System tries to authenticate a user.		
Arguments Received: **Data Type:**	**Notes:**	
String UserName	The username will be check in database for multi classes	
String Password	**.** The password will be check in database for multi classes	
Messages Sent & Arguments Passed:	**Data Type:**	**Notes:**
	History	
Arguments Return: **Data Type:**	**Notes:**	
UserID,Boolean	Boolean should be true, UserID should not be null	

Algorithm Specification:
If(username==Username in database& password== password in database)
Return true, userID
Else
Display error message		

Misc. Notes: None		

Method Name: CreateTask	**Class Name:** Task	**ID:302**
Contract ID: 102	**Programmer:** Jimma,Michael	**Date Due:** 01/05/2016
Programming Language: ***PHP** ***SQL**		
Triggers/Events: the user posts a new task In the database		
Arguments Received: **Data Type:**	**Notes:**	
String UserID		
Task TaskInfo	**.**	
Messages Sent & Arguments Passed:	**Data Type:**	**Notes:**
GetUser()	String	Will return userID and boolean
	History	
Arguments Return: **Data Type:**	**Notes:**	
void	This method will post the task in the task Database	

Algorithm Specification:

```
If(GetUser(username,password))
{
        newTask.createTask(UserID,TaskInfo)
}
```

Misc. Notes: None

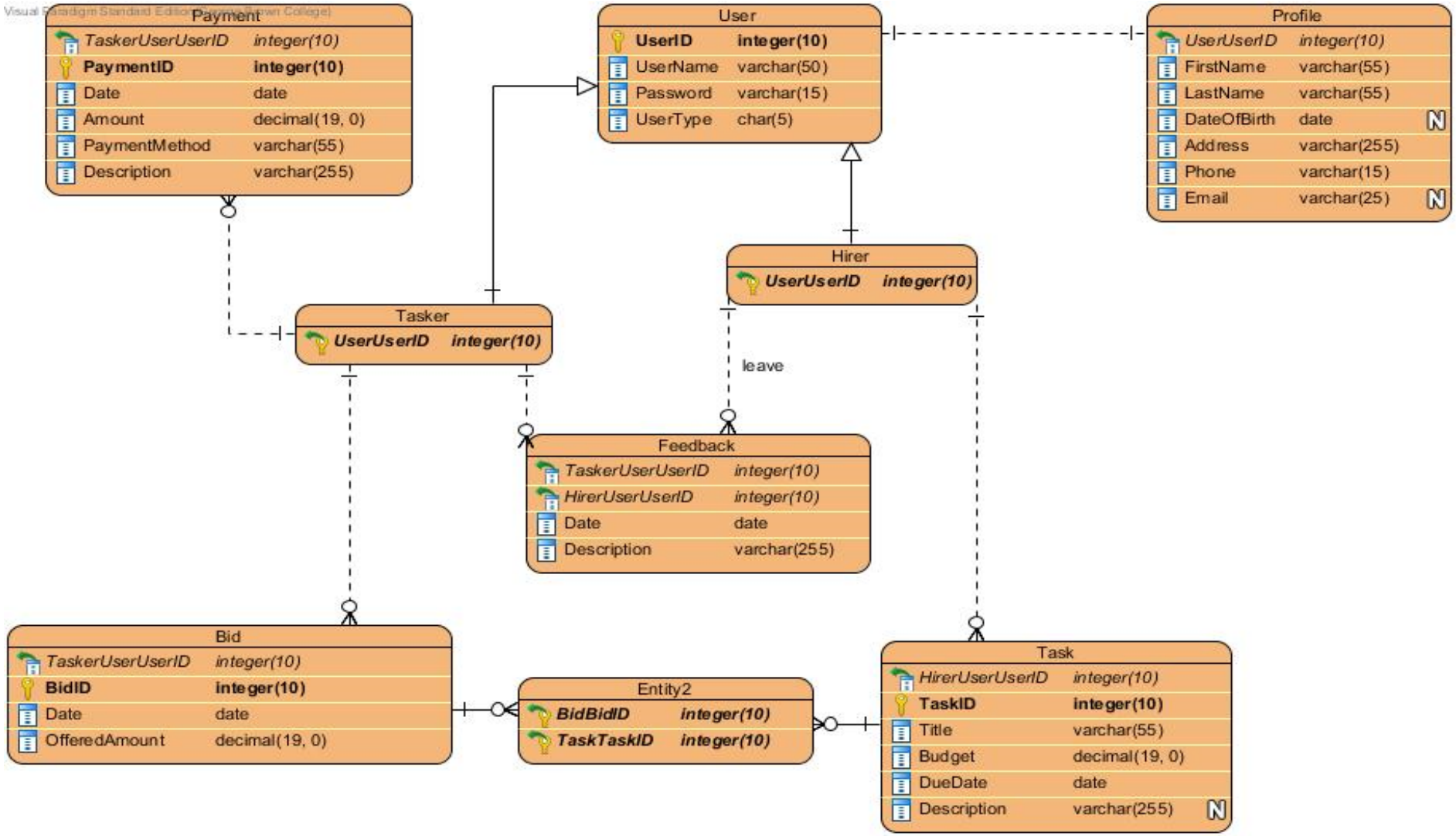
Method Name: getDetails	**Class Name:** Task	**ID:** 303
Contract ID: 104	**Programmer:** Haddad,Ramzi	**Date Due:** 1/05/2016
Programming Language: ***PHP**		
Triggers/Events: user select a task to		
Arguments Received: **Data Type:**	**Notes:**	
int TaskID		
	.	
Messages Sent & Arguments Passed:	**Data Type:**	**Notes:**
	History	
Arguments Return: **Data Type:**	**Notes:**	
Task selectedTask		

Algorithm Specification:

```
public Task Findtask(TaskID)
{
        Return taskOJ;
}
```

Misc. Notes: None

ER diagram


Payment
TaskerUserUserID integer(10)
PaymentID integer(10)
Date date
Amount decimal(19, 0)
PaymentMethod varchar(55)
Description varchar(255)
User
UserID integer(10)
UserName varchar(50)
Password varchar(15)
UserType char(5)
Profile
UserUserID integer(10)
FirstName varchar(55)
LastName varchar(55)
DateOfBirth date N
Address varchar(255)
Phone varchar(15)
Email varchar(25) N
Hirer
UserUserID integer(10)
Tasker
UserUserID integer(10)
leave
Feedback
TaskerUserUserID integer(10)
HirerUserUserID integer(10)
Date date
Description varchar(255)
Bid
TaskerUserUserID integer(10)
BidID integer(10)
Date date
OfferedAmount decimal(19, 0)
Entity2
BidBidID integer(10)
TaskTaskID integer(10)
Task
HirerUserUserID integer(10)
TaskID integer(10)
Title varchar(55)
Budget decimal(19, 0)
DueDate date
Description varchar(255) N

Data Access Speed Optimization strategies

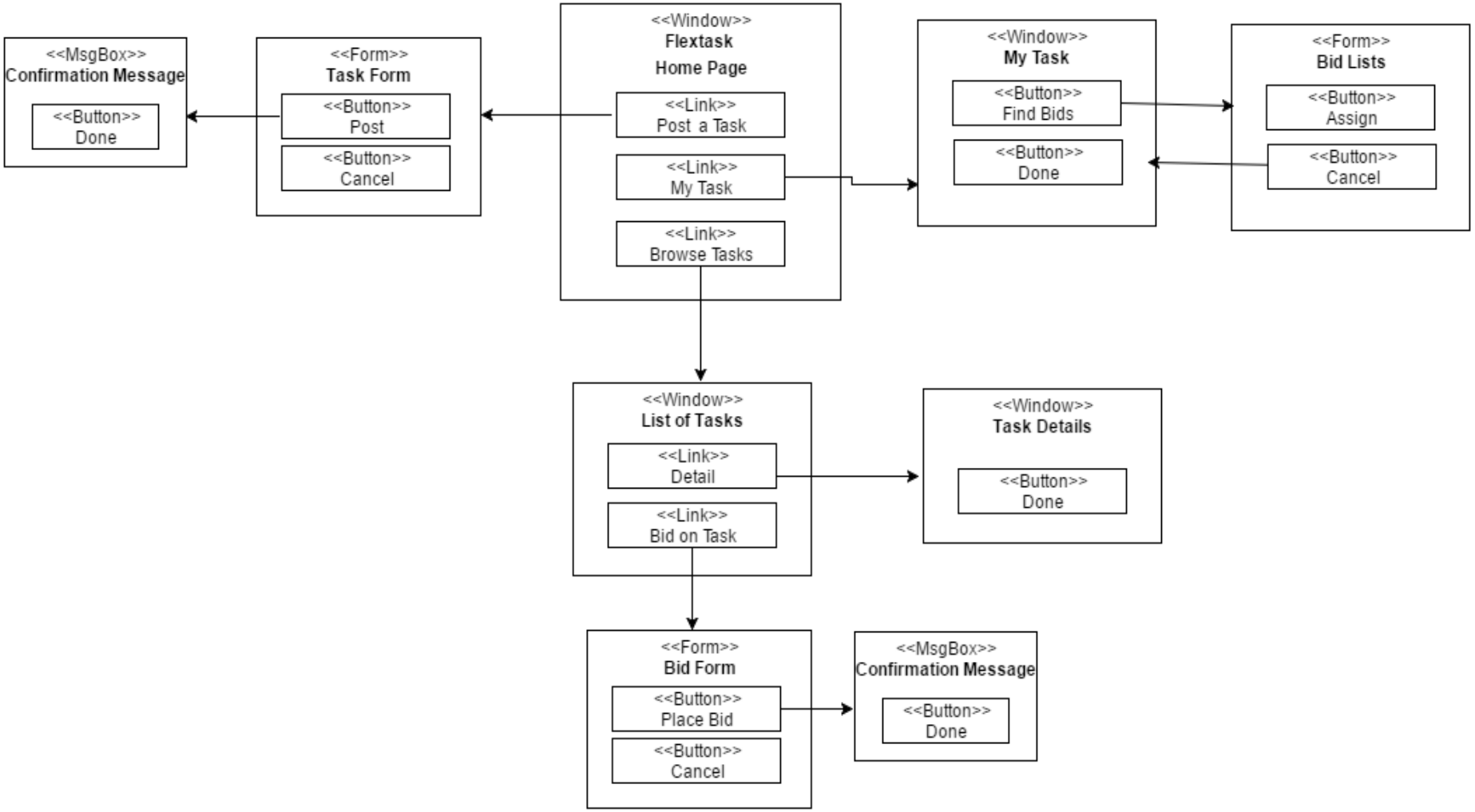
Use scenarios

Use scenario: hirer post a task in the system
1- Hirer will login in the system using his credentials(username and password). **2-** Then the hirer will navigate to the post task page. **3-** The hirer will fill form about information about the task and click post task button. **4-** The system will update the task table in the database **5-** The hirer will receive a confirmation message and will be redirected to the home page.

Use scenario: tasker bid on existing post
1. User navigate to the bid page. 2. User fill the place bid form. 3. The system will update the bids table **4.** User will be directed to home page.

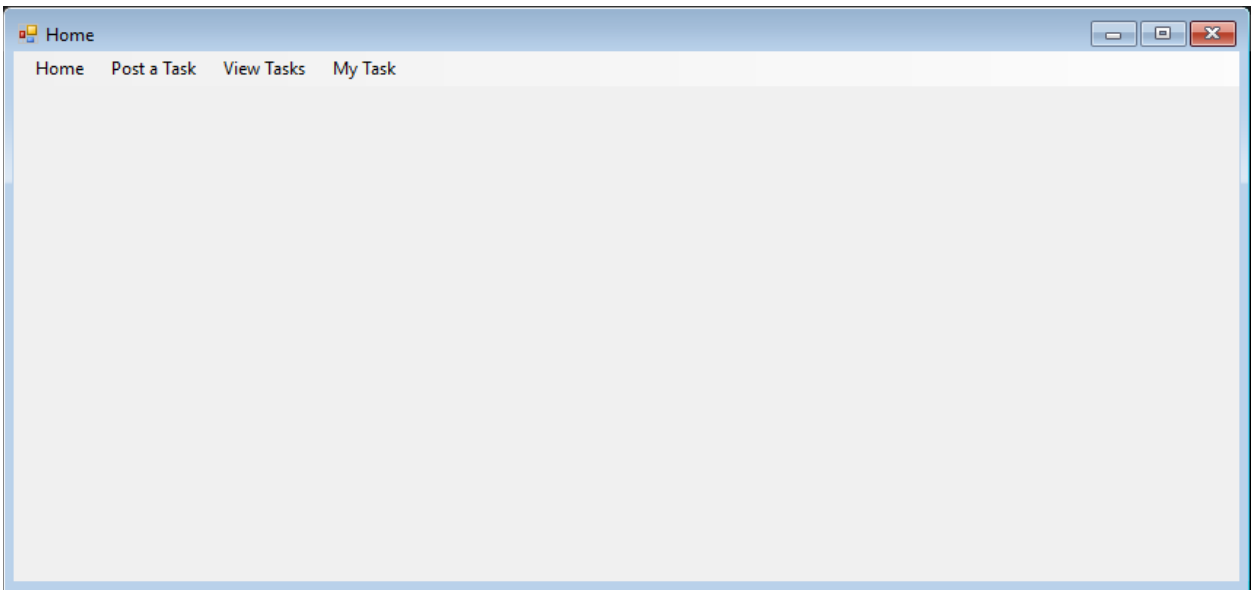
Use scenario: user want to view task
1- User navigate to the tasks page . **2-** The system will display all available posts **3-** User click on the desired object. **4-** New screen with post details will appear

Windows Navigation Diagram (WND)

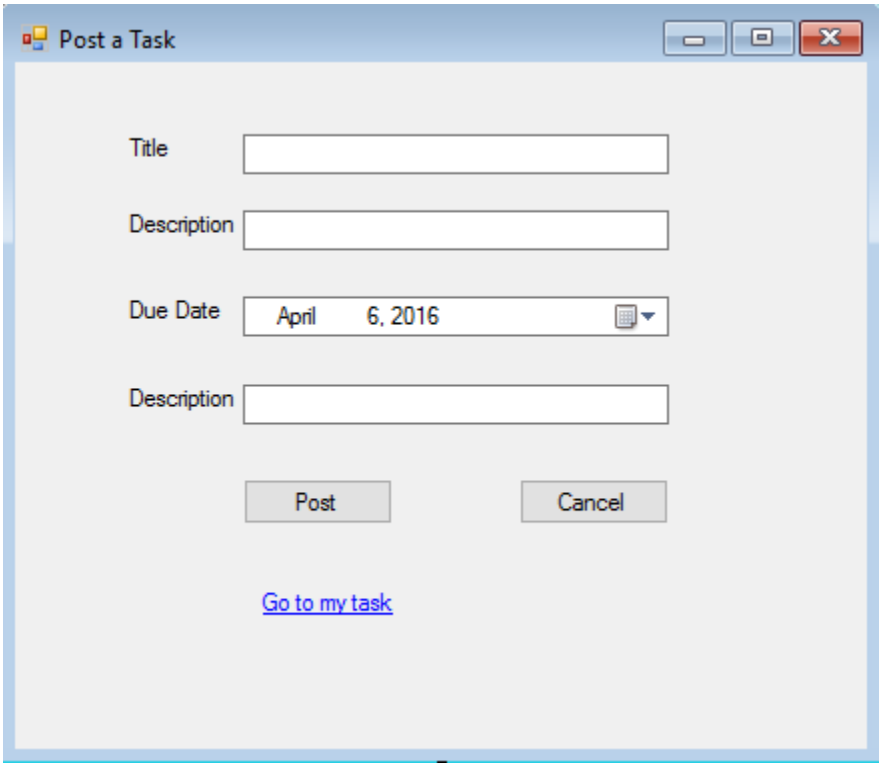


Interface Design Prototype

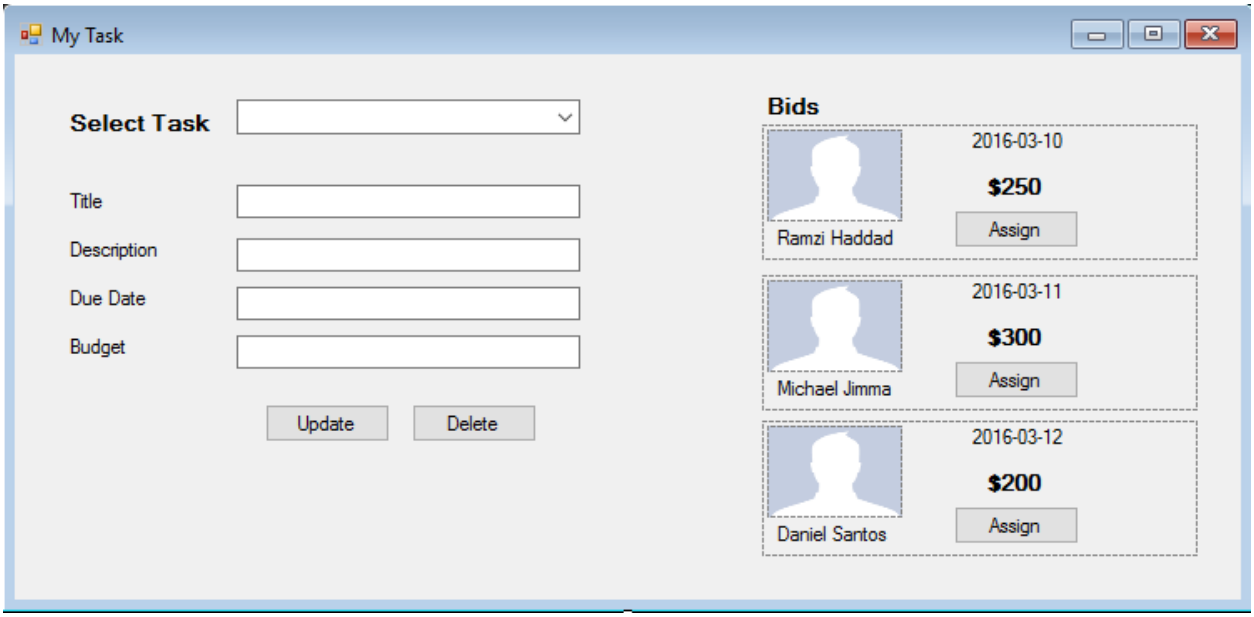
Main page



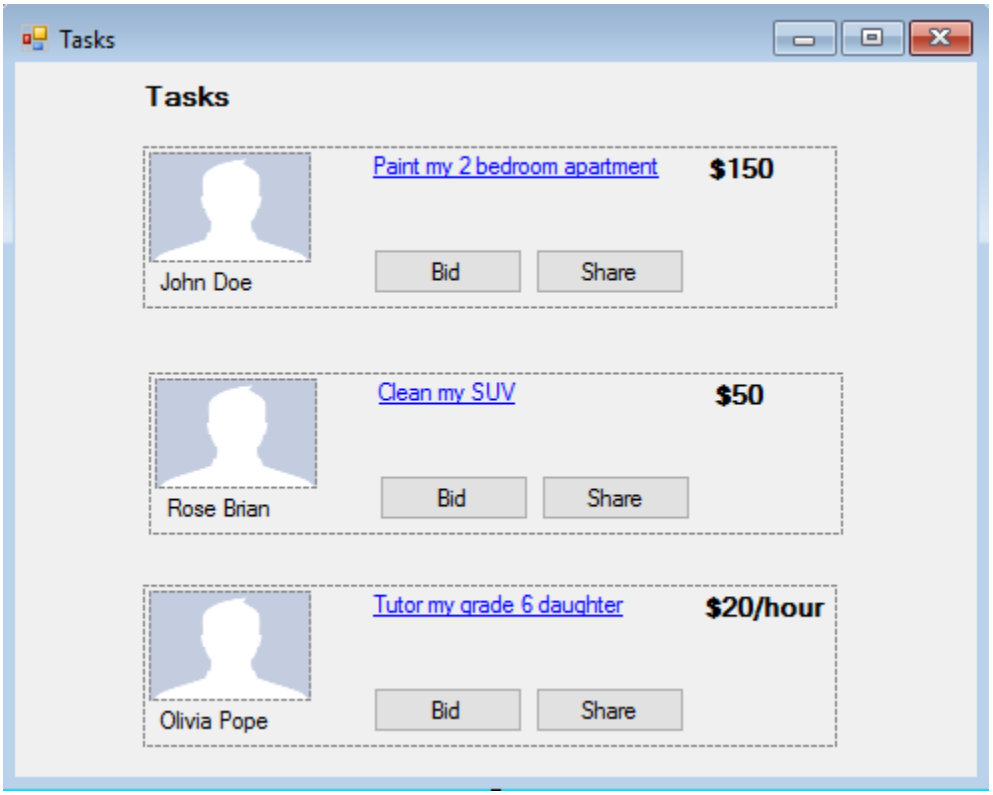
Post a task page



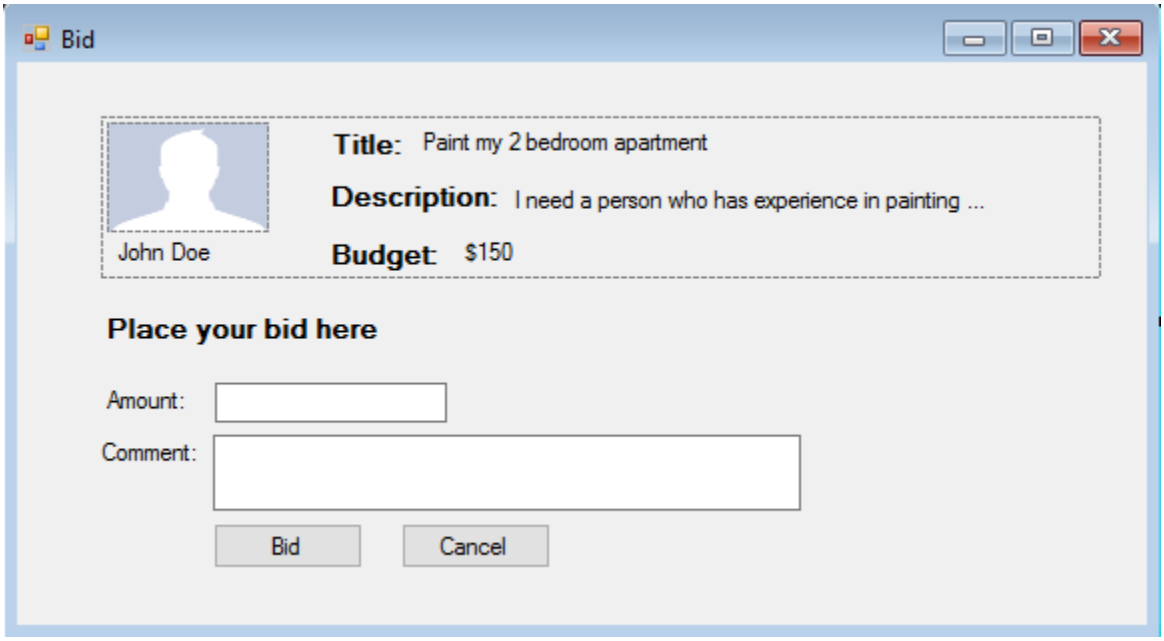
View my tasks



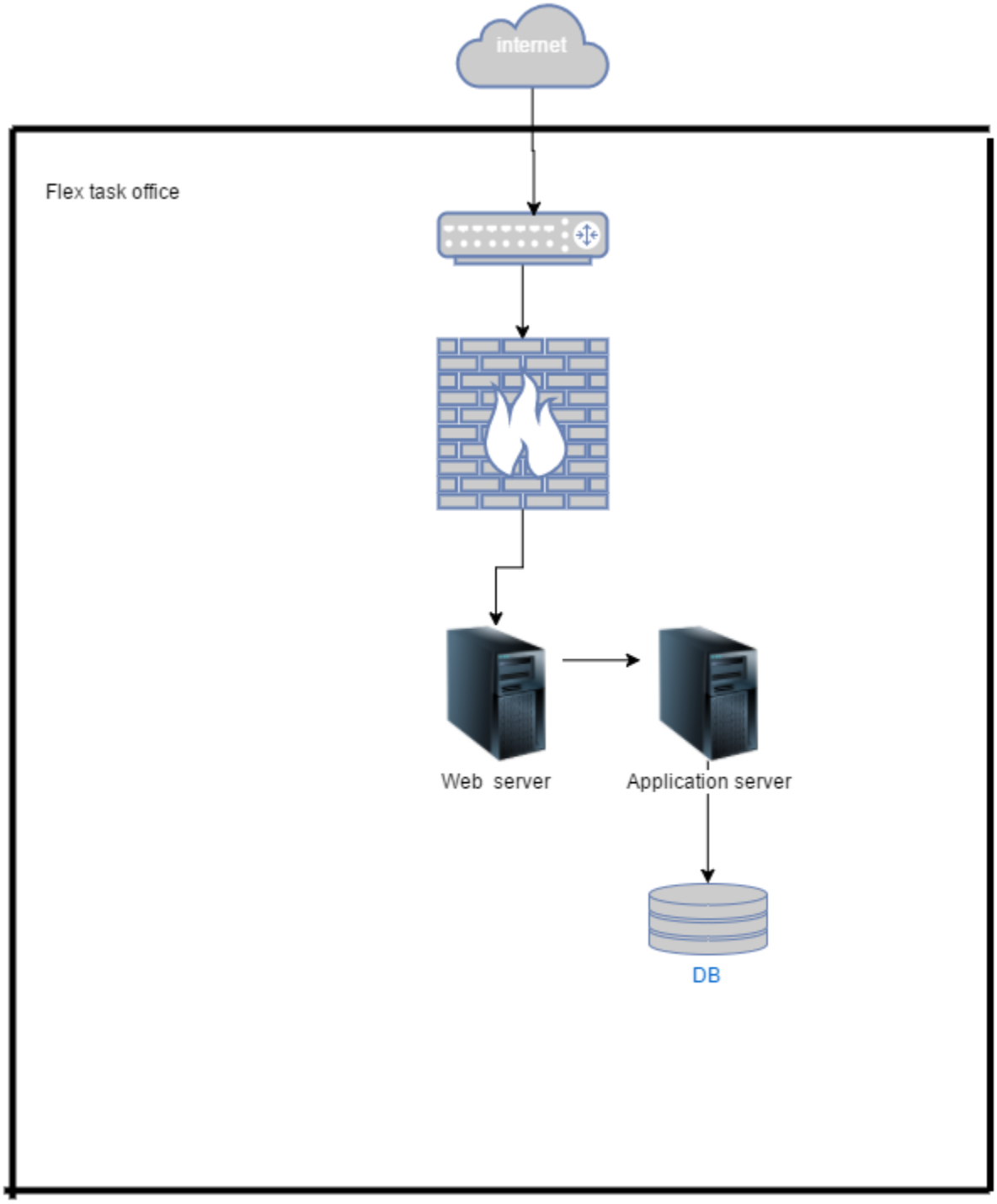
View all tasks



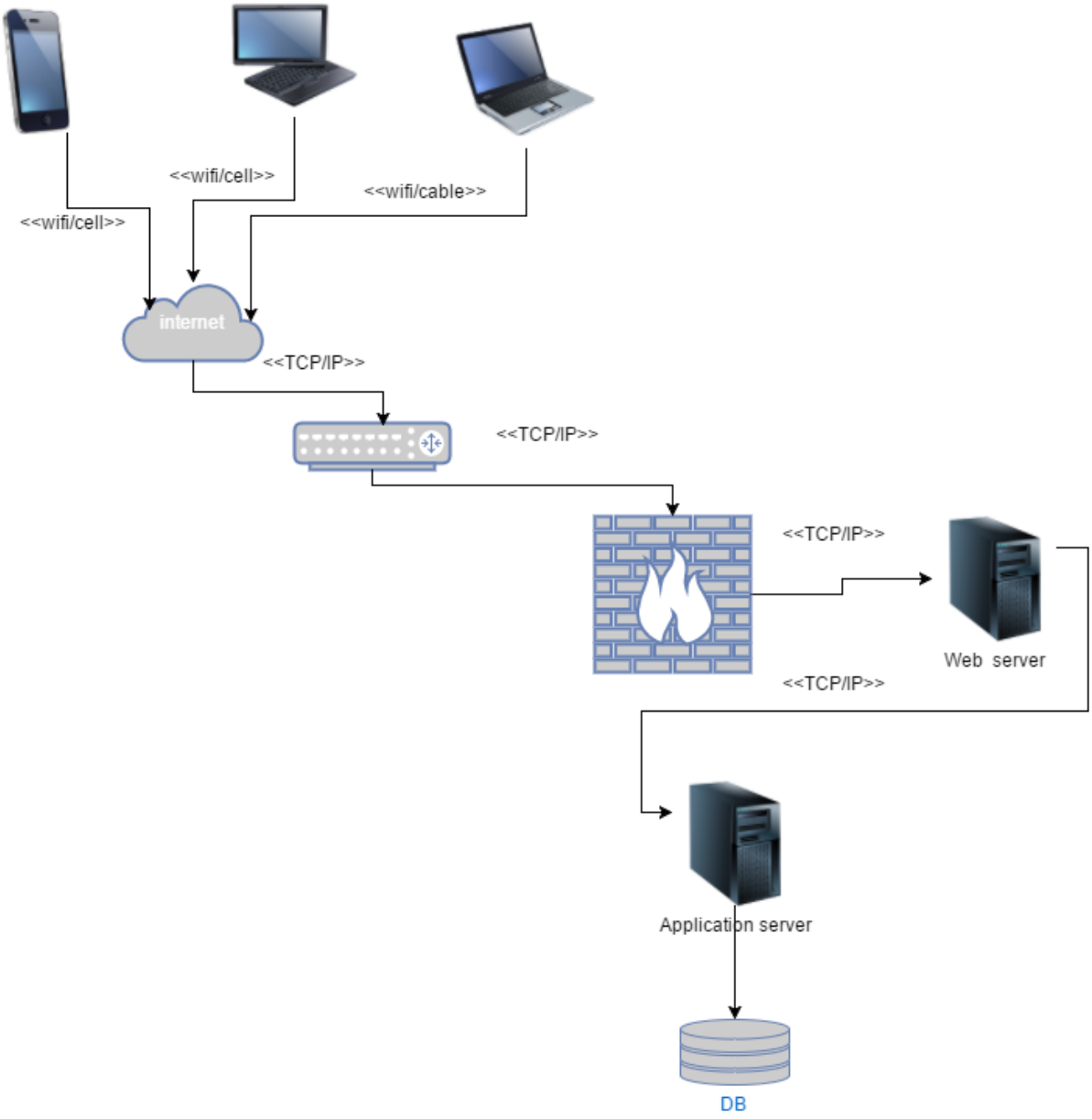
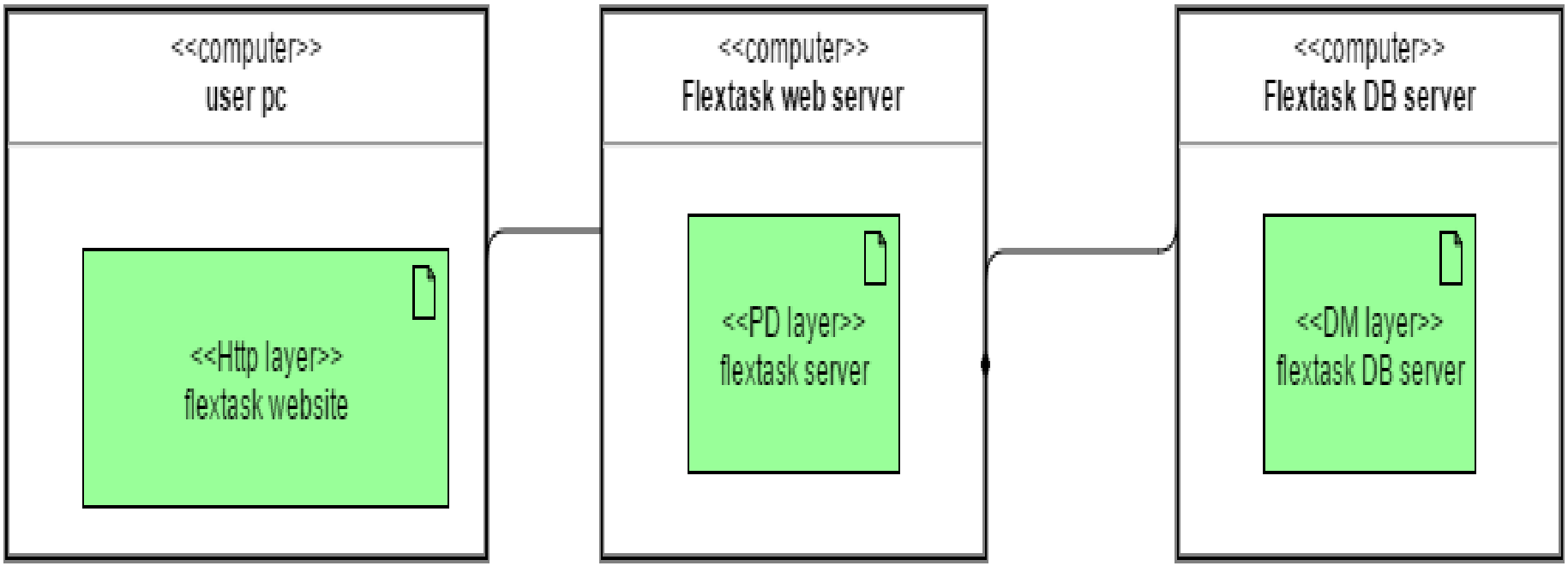
Bid on a task page



Deployment Diagram



Hardware component



Software diagram