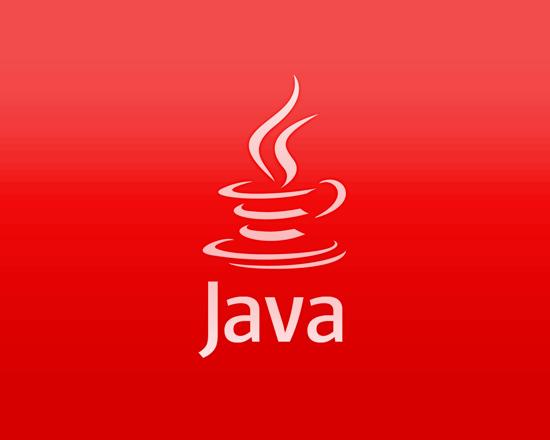
Friendship management system



Friendship management system

Submitted by

Sirat Binte Siddique (12303021)

**Department of Computer Science & Engineering at IUBAT**

**Supervised by**

Md.Khalequzzaman

Faculty of Computer science and Engineering Department

International University of Business Agriculture & Technology

**Date: 9-12-2015**

****

**International University of Business Agriculture & Technology**

**Letter of Transmittal**

Md.Khalequzzaman

Faculty of Computer science and Engineering Department  
IUBAT-International University of Business Agriculture & Technology,   
4 Embankment Drive Road, Sector-10

Dhaka-1230.

Dear Sir,

As agreed in our December 9 contract, we are submitting the attached report of project entitled “**Friendship Management System*”*** This report has been prepared pursuant to the requirements for the CSC 383: Java programming language project in spring 2015 in IUBAT. Working for this report has been interesting and informative experience for us. I learned many unidentified facts, which i believe will be supportive to omy lives and my professional career in the future.

I hope you find this report satisfactory. After that, if you have any query i will be available any time.

Sincerely yours,

Sirat Binte Siddique (12303021)

**Students Declaration**

I am declaring that this report on the topic of “**Friendship Management System**” has only been prepared for the partial fulfillment of the course requirement of CSC 383: Java programming language project. It has not been prepared for any other purpose like reward or presentation.

**Acknowledgment**

At first I would like to thank IUBAT (International University of Business Agriculture & Technology) that I am studying here & giving a lot of opportunity to do right thing. It is a great opportunity for me for giving a great place to do this work.

We also would like to thank Md.Khalequzzaman, who has been our advisor for past few months, for providing scholarly and practical advice. I am indebted to him for his patience and support, and have enjoyed sharing learning experience that has been enlightening, engaging, and gratifying. This dissertation wouldn’t have been possible without the discussion, help, support, and friendship of all of our friends, faculty, and IUBAT.

**Table of Contents**

Contents page

Executive summary………………………………………………………………………………………………….. ..2

Abstract ……………………………………………………………………………………………….……………………. .3

Objective ……………………………………………………………………………………………………………………..3

Methods ………………………………………………………………………………………………………………………4

Resource ………………………………………………………………………………………………………………………4

Schedule………………………………………………………………………………… .……………………………………4

Product breakdown…………………………………………………………………………………………………….4

Work breakdown………………………………………………………………………………………………………….5

Activity dependency-------------------------------------------------------------------------------6

Gannt chart-----------------------------------------------------------------------------------------6

System design and specification ----------------------------------------------------------------7

Use case diagram ---------------------------------------------------------------------------------7

Flow of events……………………………………………………………………………………………………………10

Sequence diagram--------------------------------------------------------------------------------12

# Executive summary

Friends are a really important part of out lives. Having people around that we can trust and rely on gets we through the bad times and makes the good times so much better. But as with any relationship, friendships require a bit of work to keep them going strong**.**So for managing purpose of all my friend information and their activity we can develop friendship management system where we put friend information like address phone number,their activity etc.

**ABSTRACT**

The friendship management system will do the following activities:

 Adding new friend in the System

 Delete friend from the System

 Change friend’ details

 Get Statement

In designing the system, Unified Modeling Language (UML) has been followed here although there are many ways to do the same. The reason for using UML is that it is very effective in object-oriented design and analysis. UML is the graphical notation to express design, and thus it is very easy to communicate with the system using such diagrams. Three different types of diagrams have been depicted here for design and specification purpose. Use case diagrams are used to show the interaction between users and the system. Class diagrams are used to show the interaction between the classes and each class has specific attributes and operations. Finally, sequence diagrams are used to show the flow of functionality of the system.

Friendship is a relationship of mutual [affection](http://en.wikipedia.org/wiki/Affection) between two or more people.[[1]](http://en.wikipedia.org/wiki/Friendship" \l "cite_note-1) Friendship is a stronger form of [interpersonal bond](http://en.wikipedia.org/wiki/Interpersonal_relationship) than an association. Sometimes we can not remind our friend and their information. So we can solve this problem through develop a friendship management system. In manual system, it is not possible to write down all the information about friend. So, like as manual system a database is required for keeping the transaction record and in this regard a MySQL database has been designed here. Designing database is not an easy task as redundancy may occur. To eliminate the redundancy the normalization process has been followed here.

In implementing the coding java programming language has been used here. Java is a class based language. Testing any system is the most important because all the works implemented is behind the testing. According to the test plan each test case has been examined whether it satisfies each test result. If the result is okay then we can conclude that the system is working perfectly.

**OBJECTIVES**

The following objectives have been set for the project:

 First of all i will get an opportunity to gather friend information knowledge in building a friendship manage system with the help of our ongoing technologies learnt from the computing course.

 The project will help us in developing problem solving skills.

 We can explore a new technology (i.e. java) that was taught in the course.

 Finally developing a good system will definitely demonstrate our technical abilities that can also be shown to potential employers or clients.

**METHODS**

The methods i will follow in completion of this project are as under:

 As the system is a friendship management application, so first of all we would like to review the friend list, database and java books for researching of the project.

 Secondly, as I am going to develop the system in context of my own friendship management environment, therefore, I intend to read manual working procedures and advantages and disadvantages of the manual systems of different management.

 I also follow some management system like faculty, student etc for reviewing requirements and design.

 Regarding design of the application, Unified Modeling Language (UML) will be used.

 The development application will be a desktop based (MYSQL will be the database whereas Java will be the programming language).

**RESOURCES**

We would like to use the following resources for successfully completion of the project:

 Details friends information.

 Java programming books.

 Computer database books.

 Follow another some management system like student management, faculty management system.

**SCHEDULE**

**Product Breakdown**

We have divided the project into the following products (component):

**Friend information System:** This is the core of friend information management functionality because the friend management activity was established on the basis of keeping information of friend in the safe place. Under this system, we will be able to keep all of our friend information.

**Report:** This will produce reports when the system needs to do so.

**Friend Management:** This will be responsible for maintaining the names of those friends who have information in this system.

**Work Breakdown**

The following under noted work breakdown activities needs to be carried out for every products of the project:

a. **Design & Specification:** Design & specification phase activity deals with modeling the component/product.

b. **Coding:** Programming of the design for the component/product carries out in this phase.

c. **Test Plans:** In this phase test plans need to be developed to test each product/component and then integration of all the products/components.

d. **Testing:** Under the test plans, code needs to be tested at each component level and also when it is integrated with other components. Test results need to be evaluated at individual component level and at the level of integration also.

In addition to the above the following activities also need to be carried out for successful completion of the project:

**a. Preliminary Investigation: -** The preliminary investigation phase deals with reviewing friendship information, networking, programming and other relevant literature. All the issues that needed for completion of the project will be adapted in this phase.

**b. Database Design: -** The friendship management application needs a database to store information.

**c. Final Report: -** All the works that have done for completion of this project is required to be documented for final report preparation and better understanding of the project.

**d.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Activity | | | | | | |
| Code | Name | Friend info system |  | Report | Friend management | Total Day |
| B1 | Preliminary investigation |  |  |  |  | 5 |
| B2 | Design and specification | 2 | 1 | 1 | 1 | 6 |
| B3 | Database design | 4 | 1 | 0 | 1 | 6 |
| B4 | Coding | 6 | 3 | 1 | 2 | 16 |
| B5 | Test plans | 1 | 1 | 0 | 1 | 3 |
| B6 | Testing | 1 | 1 | 0 | 1 | 3 |
| B7 | Final report |  |  |  |  | 2 |
|  |  |  |  | Total |  | 41 |

**Activity dependency:**

|  |  |  |  |
| --- | --- | --- | --- |
| Activity | | Duration (day) | Dependency |
| Code | Name |
| |  |  | | --- | --- | | B1 |  | |  |  | | Preliminary Investigation | 3 |  |
| B2 | Design & Specification | 6 |  |
| B3 | Database Design | 6 | B2 |
| B4 | Coding | 14 | B2 |
| B5 | Test plans | 1 | B2 |
| B6 | Testing | 1 | B3 B4 B5 |
| B7 | Final report | 2 | B1,B6 |

**GANTT CHART: Here 1colum=5 days**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Id** | **Task name** | **Duration /day** | **March 11-15** | | **March 16-20** | | **March 21-25** | | **March 26-31** | | **April 1-5** | | | **April 6-10** | | |
| **1-5** | | **6-10** | | **11-16** | | **17-21** | | **22-26** | | | **27-31** | | |
| **1** | Preliminary Investigation | **5** |  | |  | |  | |  | |  | | |  | | |
| **2** | Design & Specification | **6** |  |  | |  | |  | |  | |  | | |  | |
| **3** | Database Design | **6** |  | |  | |  | |  | |  | | |  | | |
| **4** | Coding | **12** |  | |  | |  | |  | |  | | |  | | |
| **5** | Test plans | **3** |  | |  | |  | |  | |  | |  |  | | |
| **6** | Testing | **3** |  | |  | |  | |  | |  | | |  | |  |
| **7** | Final Report | **2** |  | |  | |  | |  | |  | | |  | |  |

**System Design & Specification**

**The Unified Modeling Language (UML)**

The Unified Modeling Language (UML), which has been chosen here to design the friendship system, is the successor to the wave of object-oriented analysis and design. Although it is not a method and a modeling language, but most methods consist, at least in principle, of both a modeling language and a process. Modeling language is mainly the graphical notation that methods use to express designs, whereas process is their advice on what steps to take in doing a design. Although coding is the real point of software development, but what is the implication of diagrams as no user is going to thank for pretty pictures, and the user wants software that executes. UML consists of diagrams as well as precise description of those diagrams, but the fundamental reason for using UML is the communication. UML is used because it seems easier to communicate than alternatives. Natural language is always imprecise and makes it difficult for complex ideas. UML is such a modeling language, which can do a certain amount of precision and will never be lost in details. UML is invaluable because it helps to get an overall view of the system that aimed to develop. UML has various types of visual diagrams and each type of diagram’s functionality is different also. In this project, the following three types of diagrams have been used for designing and specification of the friendship system.

 Use case diagram

 Class diagram and

 Sequence diagram

**Use Case Diagrams of the system**

Use cases are one of the important parts of UML. It is used at the beginning of the project phase to find out the functionality of the system. It represents diagram and each diagram describes the interaction between a user and the system. Here the user can be human, other computers, hardware or a software system. The use case diagram for the friendship management system is as under:

Add friend

Delete friend

Edit friend details

Admin

**Add Friend:**

**Brief Description**

This use case allows the admin to add potential friend in the friendship management system.

**Flow of Events Basic Flow**

This use case starts when the admin need to add new friend in the system.

(a) The system request that the admin specify the function he/she would like to perform (i.e. Add friend).

(b) Once the admin provides his selection, then Add friend flow of event is executed.

**Add Friend.**

 A friend who sometimes need the manage their activity and information.

 The admin require about the terms and conditions of the same and also tells him/her to provide necessary documents to open an account.

 The friends provides or admin collect necessary documents.

 The admin then verifies the customer’s documents.

 The system presents a dialog to the admin doing the entry.

 Admin adds new customer’s information (name, address, number etc.) in the system .

 The Submit flow of event is executed.

**Submit Add Customer**

 The information is validated.

 The add customer details are saved in the system.

**Alternative Flows**

**Cancel Add friend:**

Here at any stage before the Submit flow of event is executed the admin can cancel the operation and the system is unchanged.

**Pre-Conditions**

The admin must be logged in to the System. Post-Conditions If the use case was successful, the friend is added. Otherwise system state is unchanged.

**Delete friend**:

Brief Description This use case allows the admin to delete friend from the friendship management system. When admin seems to that this friend information no need any more, under such occurrences the admin generally delete the friend from the friend list.

**Flow of Events**

**Basic Flow**

This use case starts when the admin seems to that this friend information no need any more. (a) The system request that the admin specify the function he/she would like to perform (i.e. Delete friend).

(b) Once the admin provides his selection, then Delete delete flow of event is executed.

**Delete Customer**

 when the admin seems to that this friend information no need any more (Case 1).

 If there is no need him/her for a long time, then the admin goes for deleting him/her as an member. (Case 2).

 The manager informs his/her decision regarding delete a friend from the friend list to the designated admin.

 The system presents a dialog to admin doing the entry.

 Admin deletes friend’s information (name, address, number etc.) from the system.

 The Submit flow of event is executed.

Submit Delete Customer

 The information is validated.

 The delete customer is saved in the system.

**Alternative Flows**

**Cancel Delete Customer**

Here at any stage before the Submit flow of event is executed admin can cancel the operation and the system is unchanged.

**Pre-Conditions**

The admin must be logged in to the System.

**Post-Conditions**

If the use case was successful, the friend information is deleted from the system. Otherwise system state is unchanged.

**Change Customer Details:**

**Brief Description**

This use case allows the admin to change friend’s details in the friendship management system. There is some information (like present address, telephone number, occupation etc.) that are very important for a friend to detect him/her can be changed in course of time. And after getting friend’s changed details the admin goes for updating the friend’s details.

**Flow of Events**

**Basic Flow**

This use case starts when the admin need to change his/her friend details.

(a) The system request that the admin specify the function he/she would like to perform (i.e. Change friend Details).

(b) Once the admin provides his selection, then Change friend Details flow of event is executed.

**Change friend Details**

 When the admin need to change his/her friend details (like address, occupation, telephone number etc.).

 The system presents a dialog to the admin doing the entry.

 Admin change details in the system who wants to do so.

 The Submit flow of event is executed.

**Submit Change Customer Details**

 The information is validated.

 The customer’s changed details are saved in the system.

**Alternative Flows**

**Cancel Change Customer Details**

Here at any stage before the Submit flow of event is executed the admin can cancel the operation and the system is unchanged.

**Pre-Conditions**

The admin must be logged in to the System.

**Post-Conditions**

If the use case was successful, the friend details are changed. Otherwise system state is unchanged.

**Class diagrams of the system**

Actually the class diagrams give a picture of classes within the model. In object oriented application, each class has relationships with other classes and they have attributes (member variables) and operations (member functions). And the main purpose of class diagrams are they show the interactions between classes in the system. The class diagrams for the friendship management system is as under -

**Sequence Diagrams of the system** To show the flow of functionality of a system we need to draw sequence diagrams. In depicting sequence diagram, we should be careful about that they should resemble the class diagrams. The sequence diagrams for the system are as follows:

