

HIGH LEVEL DESIGN

C-TALK

Version 1.1

Author:

Vipin Sahu(201452051)

Sai Prahlad (201452052)

November 11, 2016



Indian Institute of Information Technology Vadodara

Team members :

TEAM MEMBER	ID
Bhoopendra Singh	201452020
Shikhar Dhing	201452021
Venkata Sandeep	201452037
Anjali Kumari	201452042
Vipin Sahu	201452051
Prahlad	201452052
Sachin Jangid	201452060
Sunny Sankhlecha	201452061
Kenneth Tenny	201452066

Revision History

Version	Description	Authors	Reviewers
1.1	HIGH LEVEL DESIGN	VIPIN SAHU, SAI PRAHLAD	SHIKAR DHING, BHOOPEN- DRA SINGH

Table of Contents

1. Introduction.....	3
1.1 Purpose.....	3
1.2 Scope.....	3
1.3 Overview.....	4
2. Design Details.....	5
2.1 Main Design Features.....	5
2.2 Use Case Diagrams.....	6
2.3 System Components.....	7
3. User Interfaces.....	10
4. Module Description.....	13-14
5. References.....	15

1. Introduction

1.1. Purpose

- **Preliminary design**—In the preliminary stages of a software development, the need is to size the project and to identify those parts of the project that might be risky or time consuming.
- **Design overview**—As the project proceeds, the need is to provide an overview of how the various sub-systems and components of the system fit together.

In both cases the high-level design should be a complete view of the entire system, breaking it down into smaller parts that are more easily understood. To minimize the maintenance overhead as construction proceeds and the lower-level design is done, it is best that the high-level design is elaborated only to the degree needed to satisfy these needs.

1.2. Scope

A high-level design provides an overview of a solution, platform, system, product, service or process. Such an overview is important in a multiproject development to make sure that each supporting component design will be compatible with its neighboring designs and with the big picture.

The High-Level Document (HLD) documentation presents the structure of the system, such as the database architecture. The High-level design document uses non-technical to half technical terms which should be understandable to the administrators of the system. High level design provides a direction for the coders to proceed with guidelines for a software code structure with minimum number of bugs.

1.3. Overview

The overview of the High-Level Design will:

- Present all the design aspects and define them in detail.
- Describe the user interface that is being implemented.
- Describe the hardware and software interfaces.
- Describe the performance requirements.
- Include design features and the architecture of the project.
- List of some of the non-functional attributes are:
 - Security
 - Reliability

- Maintainability
- Reusability
- Performance

Security

Security is a major issue with respect to the private organizations in a competitive field like event management and advertising. We are issuing a unique username and password which will be provided by the admin to an employee. The employee then logs in with the provided initial username and password to his account to edit his/her details and sets up his/her account. If a user forgets his/her password, a one-time password will be sent to the registered email on providing registered username or email. This ensures security of the software.

Reliability

systems often reside in machines that are expected to run continuously for years without errors, and in some cases, recover by themselves if an error occurs. Therefore, the software is usually developed and tested more carefully than that for personal computers, and unreliable mechanical moving parts such as disk drives, switches or buttons are avoided.

A local database will be implemented in which all computers will be connected to the same network. Since no one will interrupt this server, there will be no disturbance. This ensures reliability of the software.

Maintainability

In some cases, maintainability involves a system of continuous _improvement - learning from the past in order to improve the ability to maintain systems, or improve reliability of systems based on maintenance experience.

In telecommunication and several other engineering fields, the term maintainability has the following meanings:

A characteristic of design and installation, expressed as the probability that an item will be retained in or restored to a specified condition within a given period of time, when the maintenance is performed in accordance with prescribed procedures and resources.

The ease with which maintenance of a functional_unit can be performed in accordance with prescribed requirements.

Reusability

The code written and the components used should have the ability to be reused with no problems. Code must be written with well specified comments so that whoever uses the code later can easily get the content and is able to understand the flow and functioning of the program logic. Everything is completely reusable to anyone.

Performance

Whichever the project, it is mandatory that database of that project have to be modifiable. The updating, deletion and modification to the database must be very fast so that it does not affect the performance of the product and does not give unrealistic time lags to achieve the output.

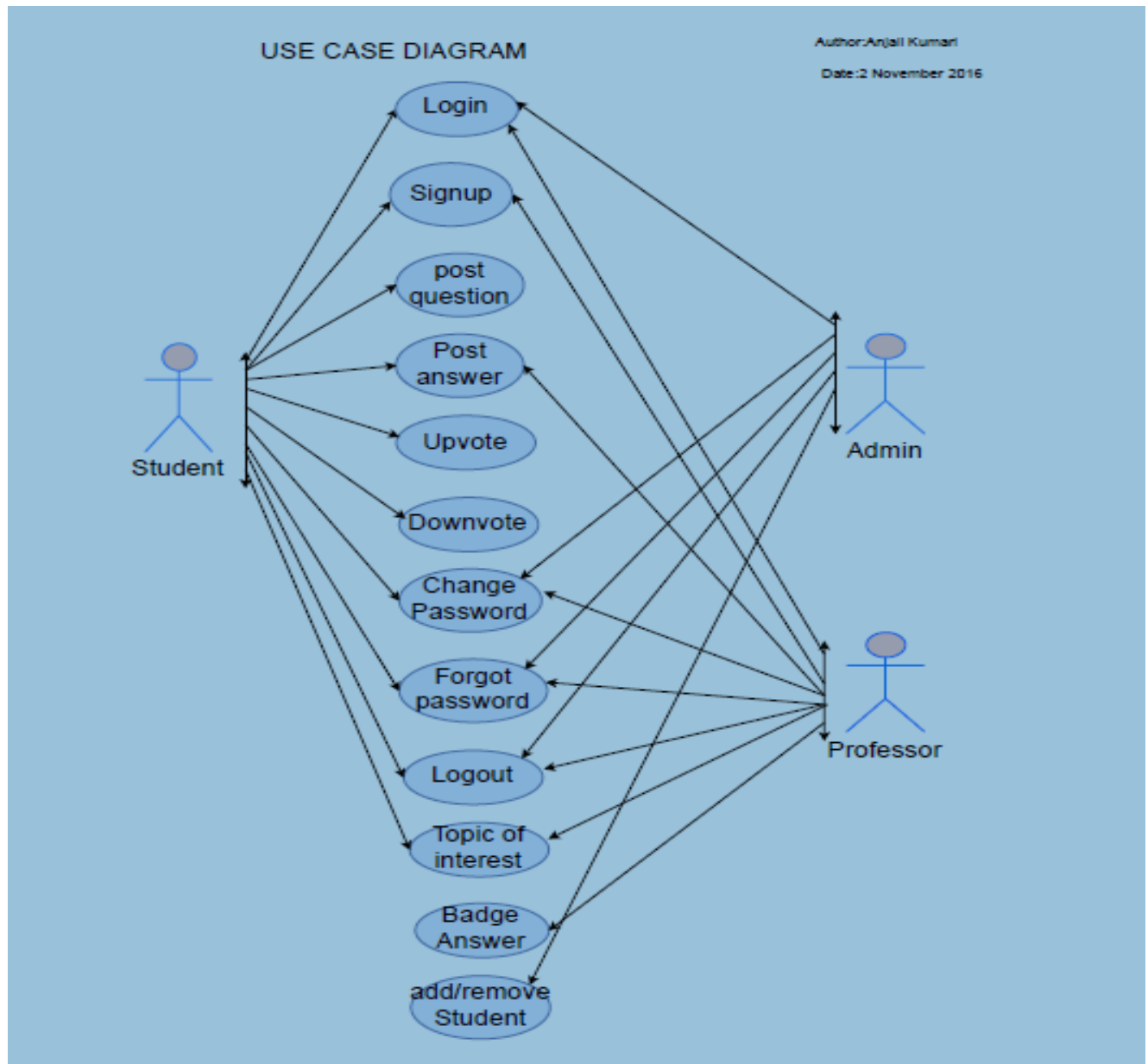
2. Design Details

2.1 Main Design Features

The main design features include five major parts: the architecture, the user interface design, external interface, the database, process relation and automation. In order to make these designs easier to understand, the design has been illustrated in below diagrams.

2.2 Use Case Diagrams

Use case diagrams give us information about the dynamic behavior of the system. The internal and external agents interacting with the system are known as actors. Every action with the system is started by these actors. Every actor has an association with one or more use cases that give information about the interaction of actors with the system.



2.3 System Components

2.3.1 Database Tables

- User_ID table

Name	Type	Nullable	Unique
User_ID	Varchar	No	Yes(Primary Key)
Name	Varchar	No	No

Password	Varchar	No	No
Type	Varchar	No	No

- **Question table**

Name	Type	Nullable	Unique
Question_ID	Varchar	No	Yes(Primary Key)
Tag	Varchar	No	No

- **Answer table**

Name	Type	Nullable	Unique
Answer_ID	Varchar	NO	Yes(Primary Key)
Question_ID	Varchar	NO	Yes
User_ID	Varchar	NO	Yes

- **TOI table**


Name	Type	Nullable	Unique
User_ID	Varchar	No	No
TOI	Varchar	No	No
User_ID, TOI	-	-	Yes

2. User Interfaces:

Basically, user interfaces mean how the user and the system interact with each other, how the it looks like, how it works, it is basically only for the user made by developer. The system backend provides a frontend for the users to give various commands to the system to get different tasks done. User friendly interfaces have become need for the day when the performance of a software is determined by user frustration. We have designed our user interfaces to be as self-explanatory and as user friendly as possible. We have tried to use the laws of human computer interaction to facilitate the interaction of user with the software in a natural and intuitive way. The prototypes of the user interfaces for our software are:

- **Admin Page**

All about Ads



Home LogOut Admin Page

User Name:

Password:

Add User

- **Login Page**

Log In

Username

Password

[Forgot Password](#)

Login

[Sign Up](#)


- **Register Page:**

Register ▾ Left 272 Top 81 Width 356 Height 619 ⓘ Offline

Register

User Name

Name

Date of Joining (dd-mm-yyyy) 


Email Address

Enter Password

Confirm Password

- **Forgot Password:**


All about Ads

Welcome XYZ! 

[Home](#) [LogOut](#) [Admin Page](#)

User ID or E-mail:

- **Change Password:**

All about Ads		Welcome XYZ!	
		Home	LogOut Admin Page
Old Password	<input type="password"/>		
New Password	<input type="password"/>		

- **Question:**

Questions		Home LogOut Admin
1	Q1	
2	Q2	
3	Q3	
4	Q4	
5	Q5	

- **Answer:**

Question_ID:-xyz	Home LogOut Admin
Question_id:-xyz question here.	
Answer_id:- all given answer of the question.	

4. Module Description

The different modules present in our software to make it a one big system are:

- **Login Module:**
The login module helps the user to login into application and access all of its functionalities. If the user enters wrong credentials, login failure will occur and the user will be prompted with wrong username or password message.
- **Registration**
If the details provided by the user is correct, then he will allow to create account. Then user can register in our web app or say we have to allow him. In registration page there are many mandatory information given by the user but we not show him in the front of other users.
- **Forgot Password (Profile)**
You can choose your topic of interest/Edit.
On entering the registered username or email id, an email consisting of new random password will be sent to the registered email. Then the user can re login to the application using the sent password. The user can change the password later on if he/she wishes to.
- **Question/Answer**

The software is specially question and answer based platform. You can ask question and you can also answer someone else question. If you want then you can up vote or down vote the answer.

5. References:

1. High level design document v1.0, Team 04, IT314 Software Engineering, Summer 2015, IIIT Vadodara.