Online Help Desk (OHD)

This project is aimed at developing an Online Help Desk (OHD) for the facilities in the campus. This is an Intranet based application that can be accessed throughout the campus. This system can be used to automate the workflow of service requests for the various facilities in the campus. This is one integrated system that covers different kinds of facilities like class-rooms, labs, hostels, mess, canteen, gymnasium, computer centre, faculty club etc. Registered users (students, faculty, lab-assistants and others) will be able to log in a request for service for any of the supported facilities. These requests will be sent to the concerned people, who are also valid users of the system, to get them resolved. There are features like email notifications/reminders, addition of a new facility to the system, report generators etc in this system.

**Functional components of the project**

Following is a list of functionalities of the system.

There are registered people in the system (students, faculty, lab-assistants and others). Some of them are responsible for maintaining the facilities (like, the lab-assistant is responsible for keeping the lab ready with all the equipment in proper condition, the students council is responsible for taking forward students’ complaints/requests to the faculty/administration etc).

There are three kinds of users for this system:

1. those who use the system to create a request (end-users)
2. those who look at the created requests and assign them to the concerned people (facility-heads)
3. those who work on the assigned requests and update the status of the same on the system (assignees)

There is also an ‘Administrator’ for doing the Admin-level functions such as creating user accounts, adding new facilities to the system etc.

1. A person should be able to
   * login to the system through the first page of the application
   * change the password after logging into the system
   * see the status of the requests created by him/her (the status could be one of unassigned/assigned/work in progress/closed/rejected)
   * see the list of requests (both open and closed) created by him/her over the past
   * create a new request by specifying the facility, the severity of the request (there may be several levels of severity defined) and a brief description of the request
   * close a request created by him/her by giving an appropriate reason
   * see the requests that are assigned to him/her by the facility-heads and update the status of requests (after working on them)
   * view the incoming requests (if he/she is a facility-head) and assign them to registered users of the system
   * get help about the OHD system on how to use the different features of the system
2. As soon as a request is created, a message will be displayed to the person who created the request and the concerned facility-head.
3. Similarly, when any status-change occurs for a request (such as the request getting completed etc), an automatic message will be updated to the person who created the request and the concerned facility-head.
4. A summary report on the requests that came in and requests that were serviced should be sent to every facility-head periodically (say, once in a month)

The following steps will be helpful to start off the project.

1. Decide on the list of facilities that would be supported and define it formally
2. Make a database of different kinds of users (End-users, Facility-heads, Assignees)
3. Create the front-page of the OHD system giving a brief description about the system and a login box
4. Create the help-pages of the system in the form of Q&A. This will help you also when implementing the system
5. Create other sub-systems like automatic notification, screens for various functions (like create\_new\_request,view\_open\_requests, forward\_new\_request\_to\_assignee etc)

**Requests/User Databases Fields Specification**

A database to store the requests should be designed. Request Id (which is created automatically by the system, as soon as a request is created) can be a key for this database. The range of valid values entered below as examples need not be taken as such. They can be modified by the team.

Another database to store the user information (such as the user name, login/password of the user, kind of user etc) should be designed as well. Following example is for the Requests database.

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Field Name** | **Range of valid values for the field** | **Remarks** |
| 1 | Request Id | Any valid 5 digit number (say, 00001 to 99999) | This is the key field of the database. This will be able to get all the details associated with the request from the database. |
| 2 | Requestor | Up to 15 characters in length. | This is the person who created the request. |
| 3 | Facility | Pre-defined set (like library, lab, hostel, mess etc) | It should be possible to include more facilities to the system, in future. |
| 4 | Request Date/Time | Any date/time format | This field will give the details about the date/time when a request was created. |
| 5 | Assignee | Up to 15 characters in length | This is the person who would be working on the request. This person is chosen by the respective facility-heads. |
| 6 | Status | Pre-defined set (like open, closed, assigned, work-in-progress, need-more-info, rejected etc) | This field is updated by several people (facility-head updates it to ‘assigned’ once he assigns it to somebody, the assignee updates it to work-in-progress or need-more-info (in case he needs more info to work on the request) or rejected (in case, it is an invalid request). |
| 6 | Remarks | Up to 100 characters in length | Short remark on the request. Can be updated by the End-user/Assignee/ facility-heads. This field can be increased in size, if needed. |

Note:-student can add more tables as per the requirement.