**Software Requirement Specification**

**for**

**Online Help Desk and Grievance Handling System**

**<Names>**

**Table of Contents**

Updates

1. Introduction

1.1. Purpose

1.2. Scope

1.3. Overview

2. Overall Description

2.1. Product Perspective

2.2. Functions of the System

2.3. User Characteristics

2.4. Operating Environment

2.5. Constraints

3. Specific Requirements

3.1. External Interface Requirements

3.1.1. External Services Synchronization Interface

3.1.2. Command Line Support

3.1.3. Project/Workspace Explorer

3.2. Non-functional Requirements

3.3.1. Performance Requirements

3.3.2. Security Requirements

3.3.3. Portability Requirements

**Revision History**

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Reason for Change** |
| Version 1 |  | Original Copy |

1. **Introduction**

This document is a Software Requirement Specification (SRS) for the “Online Help Desk and Grievance Handling System” project. This is the initial draft for the SRS, and it will be used for the extensions. This document is prepared by following IEEE conventions for Software Requirement Specification.

Online Help Desk and Grievance Handling System is a web application aimed at providing different services of the college to the students for which they normally have to wait too long. This application can be accessed throughout the campus. The present system which is totally manual consumes a lot of time for a request to be satisfied. Online Help Desk and Grievance Handling System will allow students to easily request for different services via this online portal. The corresponding facility heads will then take necessary actions and try to fulfil the requests. It supports three types of users:

1. **Students:**

These are the users who will request for the different facilities of the campus.

1. **Facility Heads:**

These are the users who will receive the requests and try to resolve them.

1. **Administrator:**

The Administrator will be able to add more students and facility heads to the system.

* 1. **Purpose**

The aim of this document is to specify complete description of the Online Help Desk and Grievance Handling System to be developed. It is basis for agreement between suppliers and customers about the product to be developed. Through this document, the workload needed for development, validation and verification will ease. To be specific, this document is going to describe functionality, external interfaces, performance, attributes and the design constraints of the system which is going to be developed. Therefore, intended reader groups for this software requirement specification are customers, suppliers and users.

* 1. **Scope**

This project is intended for making use of already existing technology Html and javascript. Currently there are lots of facilities provided by our College to students and faculties which are not yet online. Usually it is provided offline. However, in this connected world there are two problems with offline process. Firstly, physical presence is a must to avail these facilities and secondly, use of resources like paper, pen etc.

By moving it online, this project will remove the requirement for resources and physical presence will be avoided.

* 1. **Overview**

We are going to focus on describing the system in terms of product

perspective, product functions, user characteristics, assumptions and dependencies on the following section of this document. Next, we will address specific requirements of the system, which will enclose external interface requirements, requirements of the system, performance requirements, and other requirements.

1. **Overall Description**

This section gives background information about specific requirements of the Online Help Desk and Grievance Handling System software to be developed in brief. Although we will not describe every requirement in detail, this section will describe the factors that affect the final product.

* 1. **Product Perspective**

This software product is eventually intended for the students and faculties.

Product will be deployed to website and all users of the product will access by use of the website. Website will be main user interface where users can operate all the provided functionality. However, this website will be only a part of this organization. The user data is kept in local storage and all the execution is done. Website will only be the interface for the user data and the execution of provided functionalities.

To use this product, users are required to register through the web interface. Whenever a new user is registered, all the required data will be created in the database and a user will be given access to the portal. Later, users will be able to login and logout from the website anytime they want.

From Admin point of view, Admin will have a functionality to delete or create new users, admin will be able to reject any grievance raised.

* 1. **Functions of the System**

The different users of the system viz. students, administrator and facility heads carry out different activities. Use case diagrams provide an insight into specifying the roles the users perform.

Use case diagrams are usually referred to as behaviour diagrams used to describe a set of actions that some system or systems should or can perform in collaboration with one or more external users of the system. Each use case should provide some observable and valuable result to the actors or other stakeholders of the system.

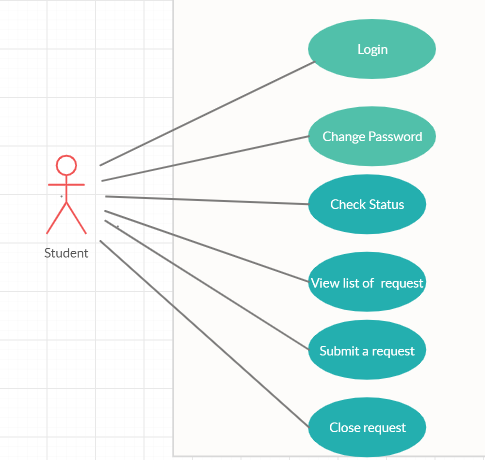


Fig. 1 Use-Case diagram of Student

Figure 1 specifies the activities that a student can perform, of which viewing a list of requests he can make and submitting a request are the major one. Moreover, he has options to change password and check status of his request.

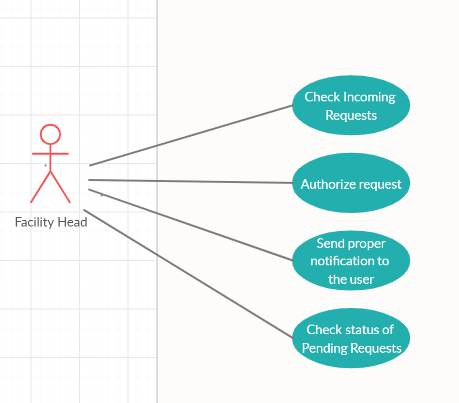


Fig. 2: Use-Case diagram of Facility Head

Figure 2 specifies the tasks that a facility head must perform such as checking the requests from the students and authorizing them i.e. taking necessary actions on them. Furthermore, he has the option to send a proper notification to the students.

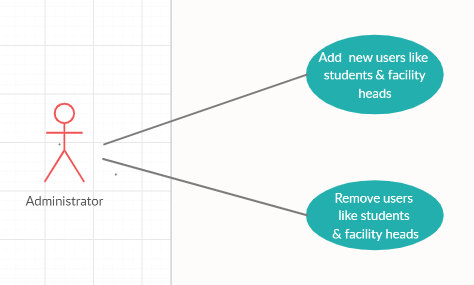


Fig. 3: Use-Case diagram of Administrator

Fig.3 specifies the job of administrator, wherein he has the ability to add or remove students or facility heads from the system.

* 1. **User Characteristics**

Users of this website will mainly be students and faculties of our college. Since it is reasonable to assume that an average user has knowledge about functionalities and usage of any website, we assume that our users will already be informed about basic functionality of the website. Also, clear documentation and tutorials about the website feature will be provided.

* 1. **Operating Environment**

This application tuns on Windows OS until it is deployed in any server. The User Interface is developed using Notepad++, HTML5, Javascript, CSS, NodeJS and MySQL.

**System Requirements**

**Operating System:** Windows 7 or above

**RAM:** 4GB or above

**Disk Space:** 500 MB Available

* 1. **Constraints**

Developers of this website should be aware that main feature of the intended website is to provide most of the college facilities available online.

Developers should also be careful about the privacy of users.

Since this website will be hosted locally and no data will be available to anyone.

1. **Specific Requirements**

With this section and later, we will describe the requirements of the software in detail. Basically, we will categorize requirements in 3 which are namely external interface requirements, functional requirements and non-functional requirements. Except non-functional requirements, requirements of the product will be detailed under this section with brief information and later sample input-output sequence and low of events will be given.

* 1. **External Interface Requirements**

In this sub section, we will describe the external interface requirements of the product in four categories which are Login Page, Admin Page, Student Page and Facility Head page.

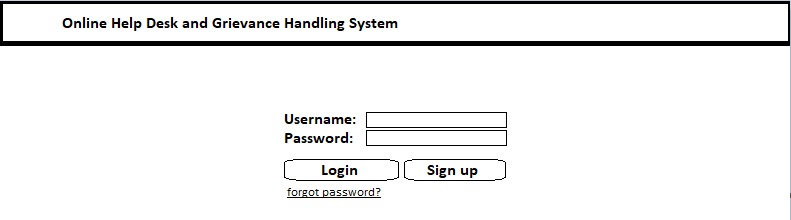
Login page will be the main landing page where all users needs to provide their credentials (if generated) or need to sign up for the first time.

Admin Page will be the landing page for logged in User who is Admin of this portal, only one admin should exist for this portal.

Student Page will be the landing page for logged in user who is student of the organization, in this page students should be able to see all request or grievance raised by them, they can complete any request raised by them by providing valid status.

Facility Head Page will be the landing page for Facility Head of the organization, in this page all request and grievance will be visible to the responsible person. They can give resolution to the requests and send it back to the corresponding student who has raised it.

* + 1. **Login Page**

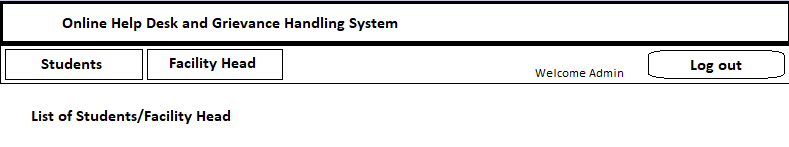
****

Flow of Events:

1. User opens Login page.
2. User tries to login to the system with his credentials.
3. System validates the provided information.
4. User Logged into the system.

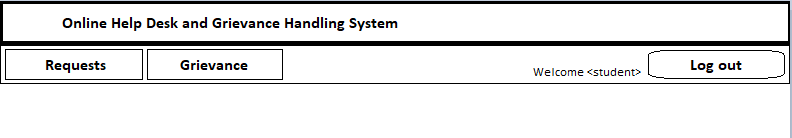
Alternative Event Flow:

1. User cannot log into the system due to incorrect credentials.
   * 1. **Admin Page**

****

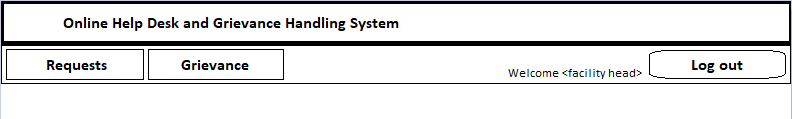
Flow of Events:

1. User Logged in as Admin will be shown Admin Dashboard.
2. 3 buttons will be shown to admin on top of the website.
3. Admin clicks on Students button
   1. List of students will be shown.
   2. Delete button will be shown against each student.
   3. Admin can click on Delete button to remove student from list.
4. Admin clicks on Facility Head Button
   1. List of Facility Heads will be shown.
   2. Delete button will be shown against each Facility Head.
   3. Admin can click on Delete button to remove Facility Head from list.
5. Admin clicks on Logout button will be logged out from the portal.
   * 1. **Students Page**

****

Flow of Events:

1. User Logged in as Student will be shown Students portal.
2. 3 buttons will be shown to student on top of the website.
3. Student clicks on Requests button
   1. List of Requests raised by that student will be shown.
   2. Student can view list of active and completed requests.
   3. Student can also create new request by filling the form available on Create new Request button.
4. Student clicks on Grievance
   1. List of Active, abandoned and solved Grievance will be shown.
   2. Student can Raise Grievance to respective facility head by clicking on Raise Grievance button.
5. On clicking on Logout button student will be successfully logged out.
   * 1. **Facility Head Page**

****

Flow of Events:

1. User logged in as Facility Head will be shown Facility Head portal.
2. 3 buttons will be shown to Facility Head on top of the website.
3. Facility Head clicks on Requests button
   1. List of Requests raised by all students for this facility will be shown.
   2. Facility Head can Resolve the Request or Mark it as on Hold by mentioning proper reason.
4. Facility Head clicks on Grievance button
   1. List of Grievances raised for this facility will be shown.
   2. Facility Head can provide proper resolution and update the grievance.
5. Facility Head clicks on Logout button will be safely logged out of the portal.
   1. **Functional Requirements.**
   2. **Non-Functional Requirements.**