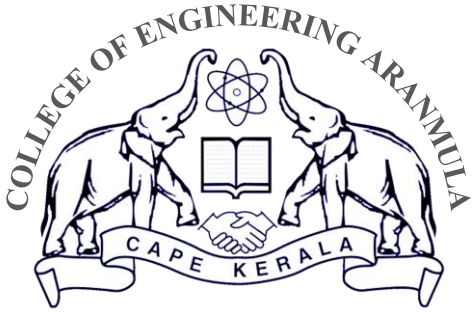


DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

MINI PROJECT

Software Requirements Specification

for

Hack4innovation,

Release 1.0

Prepared by Group A

(Adwaith4 Solutions)

1.0. Introduction

1.1. Purpose

The purpose of this document is to present a detailed description of our mini project ***Hack4innovation***. This document will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system and will be submitted to Cochin University of Science and Technology (CUSAT) & College of Engineering, Aranmula (AEC) for the award of B-Tech Degree in Computer Science and Engineering.

1.2. Scope of Project

The main objective of this project is to familiarize the college students with the concept of white hat hacking and modern hardware (IOT/VR/AR devices).

Problem definition

Many project heads don't opt for Computer science & Engineering graduates since:

- Many University Computer Science departments are in miserable shape: 10 years behind in a field that changes every 10 minutes.
- Computer science departments prepare their students for academic or research careers and refuse jobs that actually pay money.
- They teach students how to design an operating system, but not how to work with a real, live development team.

Solution

Hence there is a need of a revolutionary web app that solves all the above problems. From where we ideated the concept of ***Hack4innovation,*** focusing on following areas**.**

- White hat or Ethical hack training for familiarizing with security issues in many Indian web or mobile apps.

- Latest hardware training to students for free.
- Information about various KSUM, IEDC activities to the students.
- Registration to live events of ***Hack4innovation*** to encourage more students
- Placement & internship portal.

1.3. References

[1] Fundamentals of database systems by Ramez Elmasri & Shamkant B. Navathe

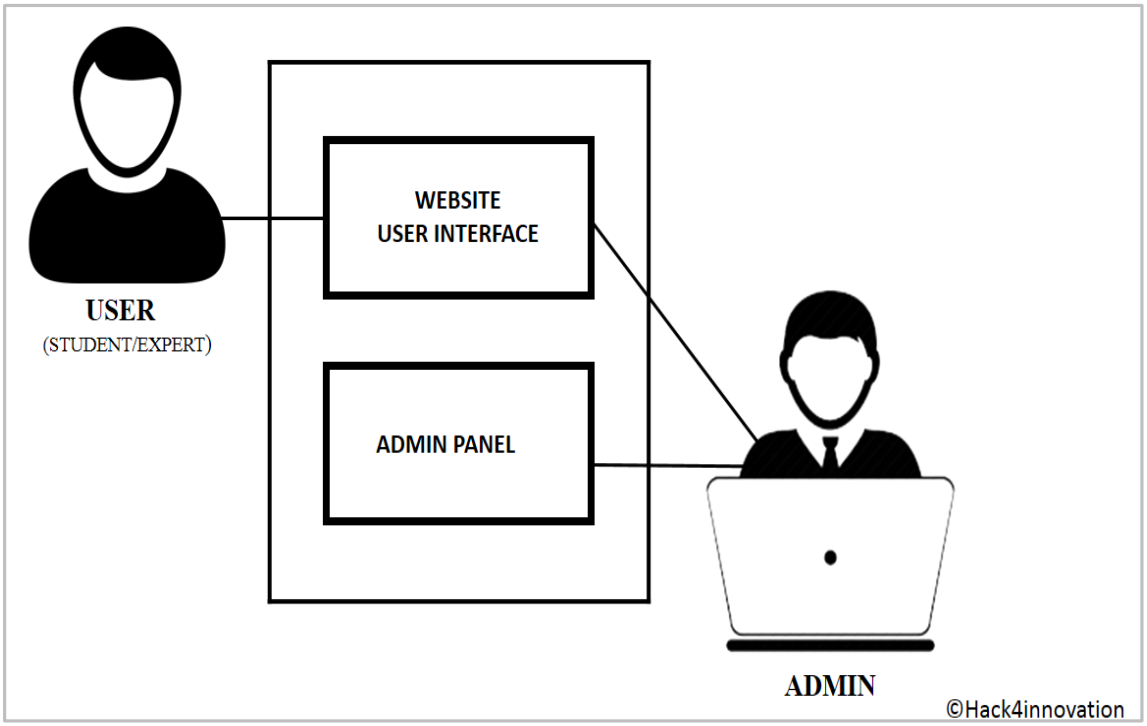
[2] Introduction to Modern Application Development -NPTEL

1.4. Overview of Document

The next chapter, the Overall Description section, of this document gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter.

The third chapter, Requirements Specification section, of this document is written primarily for the developers and describes in technical terms the details of the functionality of the product. Both sections of the document describe the same software product in its entirety, but are intended for different audiences and thus use different language.

2.0. Overall Description

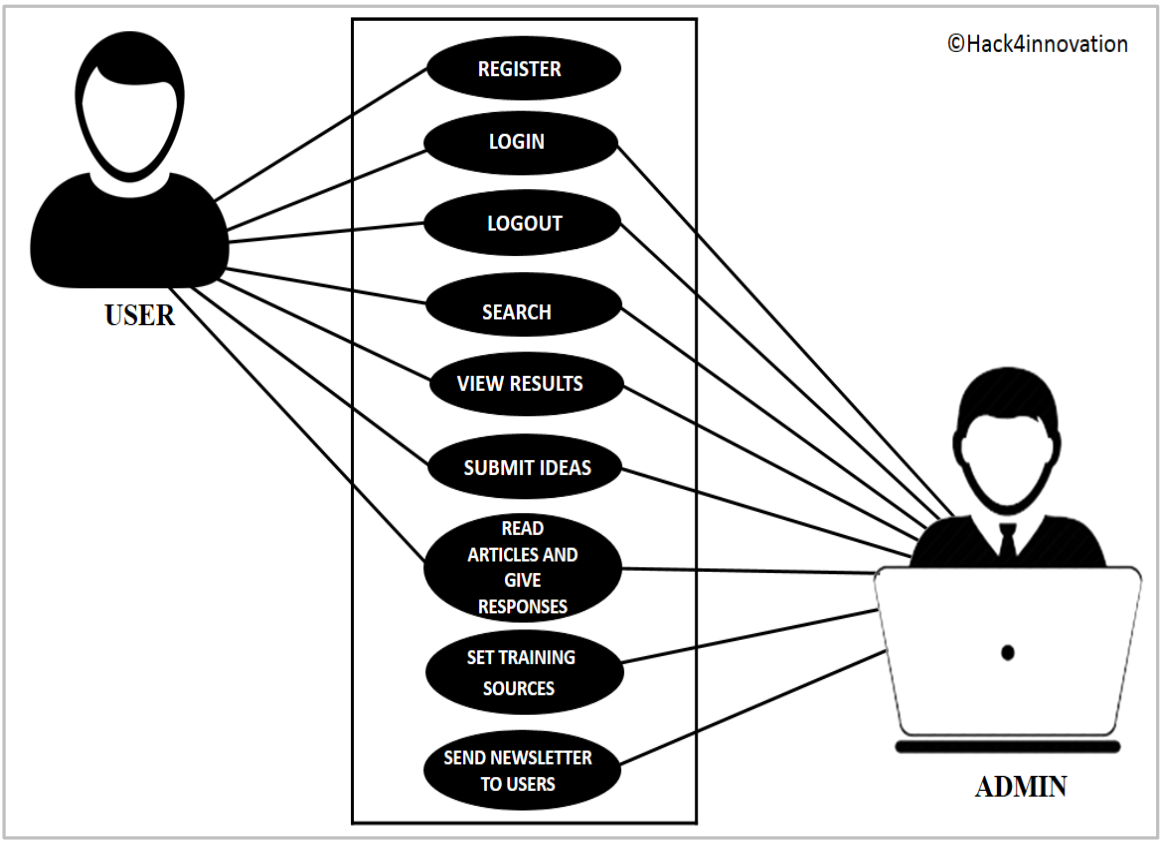


2.1 System Environment

The Hack4innovation System has mainly two active actors and one cooperating system. The User & Admin are the actors in this system. The user can communicate with the system via the User Interface of Website. The user can interact with website such as searching topics, submitting ideas or projects, like or comment on idea. The Admin Panel is controlled by the admins, who sets up the various training sources, monitors all the user activities, can use all the features of a user and is responsible for dynamicity(changes in content) of website .

2.2 Functional Requirements Specification

This section outlines the use cases for each of the actors separately. The user is the main actor in this system. A user can be either a student or a expert. The Admin plays some vital role in setting system parameters / configuration to increase the efficiency of this Hack4innovation.



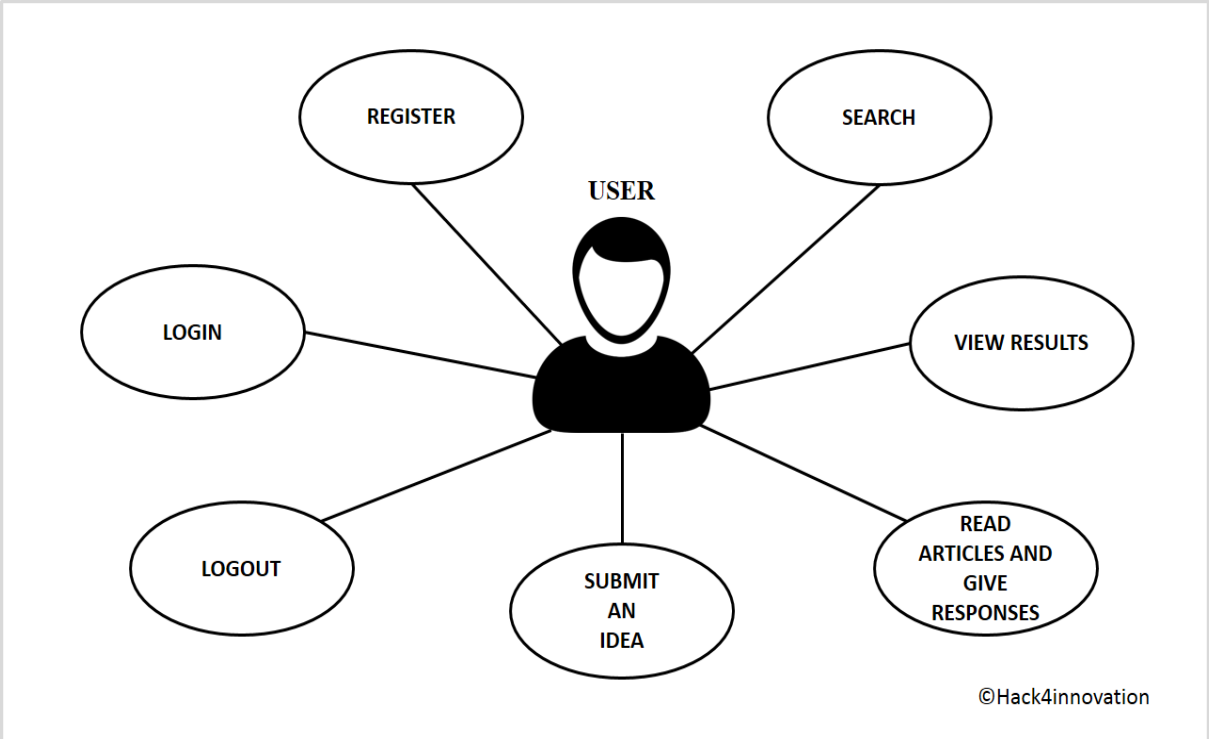
2.2.1 User Use Case

The user has the following set of use cases.

- Register
- Login
- Logout

- Submit an idea
- Search
- View Results
- Read articles & Give responses

The above stated uses cases are briefed in the following sessions.



❖ Use case: **Register**

Brief Description

The User may create an account to unlock more features of Hack4innovation website .

Initial Step-By-Step Description

Before this use case can be initiated, the User has to access the User Interface of this particular software application.

1. The user clicks on the 'Signup' button present on home page.
2. The system loads a new web page with many textboxes to be filled by user about his her details (Email id, Password, Name, Contact Number are mandatory fields).
3. The User clicks on 'Register' button.
4. The user account is only created when system verifies that user details was valid otherwise the page is reloaded.
5. After successful registration, the system would assign a unique user id & return back to home page.

❖ Use case: **Login**

Brief Description

The User may login to unlock more features of Hack4innovation website such as to submit ideas, give feedbacks to articles etc.

Initial Step-By-Step Description

Before this use case can be initiated, the User has to access the User Interface of this particular website.

1. The user inputs the registered email id and password in textboxes, then click on 'Login' button.
2. The system will verify the authentication, if successful load another page, containing various topics of article. Failure of authentication may alert user about incorrect email id or password & ask them to retry.

❖ Use case: **Logout**

Brief Description

The User may logout to clear all credentials stored in browser and exit the system.

Initial Step-By-Step Description

Before this use case can be initiated, the User must be already logged in.

1. The user panel may consist of a 'Logout' button,
2. The system will return back to the home page and clears all temporarily saved data in web browser by user's activities. (clears all session id, cookies etc.).

❖ Use case: **Search**

Brief Description

The User can initiate search for a topic, various projects and ideas.

Initial Step-By-Step Description

Before this use case can be initiated, the User has to access the User Interface of this particular software application.

1. The user inputs a topic in textbox & initiates the search by pressing the Search button.
2. The system displays a list of training topics, ideas or projects related to input topic.
3. The user selects his own requirements or interested field.
4. The system replies with the required results.

❖ Use case: **View Results**

Brief Description

This allows the user to view the search results. After initiating a search, the user is taken to another page which displays the list of phones that suits his/her need.

Initial Step-By-Step Description

Before this use case can be initiated, the User have to initiate a search via the User. Initiating a search is well explained in the section *2.2.1 Use case: Search*

1. The user search for topic based on his/her requirements.
2. The system displays a list of related topics that suits the user's need.
3. The user selects a topic of his/her choice.
4. The user is now taken to another page containing the description about the topic.
5. The user can now read the article and may give response by giving a like or comment to topic if user logged in.

❖ Use case: **Read Articles and Giving Response**

Brief Description

This allows the user to read various articles related to white hat hacking, modern hardware, ideas, projects and events to be held .The user must be logged in before giving responses to the particular articles.

Initial Step-By-Step Description

Before this use case can be initiated, the User have to select a topic either by searching a particular topic or by just visiting the popular articles.

1. The user selected a topic.
2. The user is now taken to another page containing the brief description of the topic.
3. The user can now read the article.
4. The user can appreciate the article's work by clicking on like and may give suggestions or feedback to the article on ideas, projects to improve it.

❖ Use case: **Submit ideas**

Brief Description

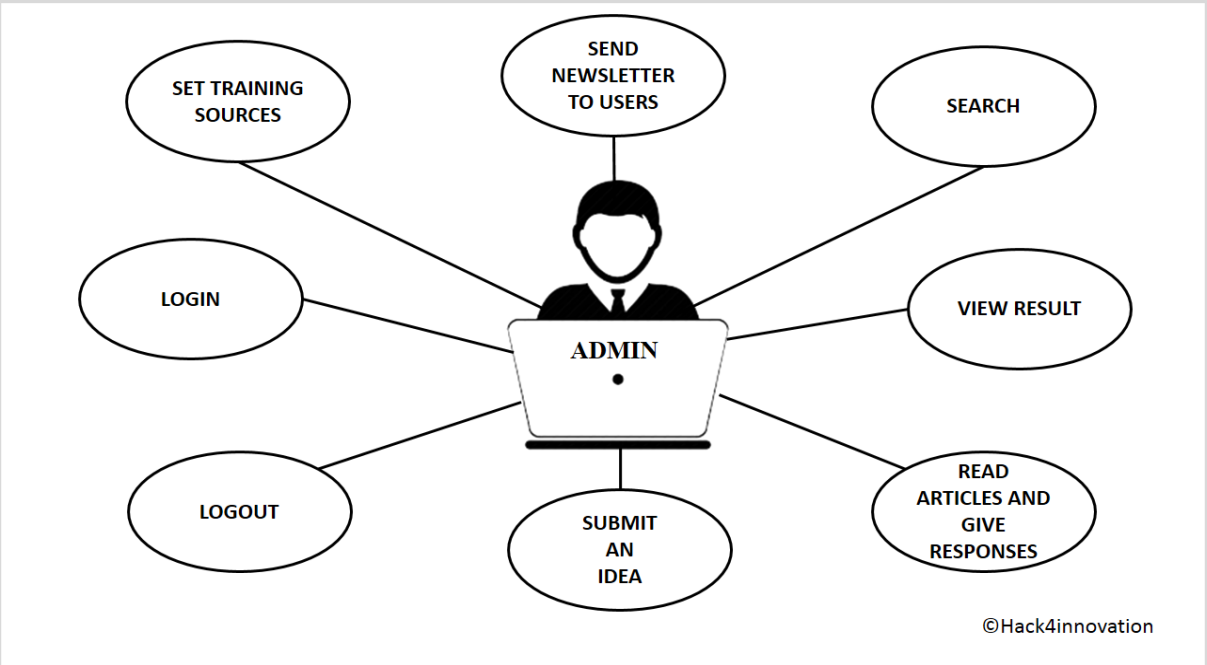
This allows the user to write various articles related to their various new ideas for projects or even submit the detail about the project they researched.

Initial Step-By-Step Description

Before this use case can be initiated, the user must be logged in before submitting idea.

1. The user clicks on 'Submit an Idea' button.
2. The user is now taken to another page containing the textbox to input details about the idea.
3. The user, then press 'Submit' button & returns to home page.
4. The admin may make modification for quality of the idea submitted by user.

1.2.2 Admin Use Case



❖ Use case: **Send Newsletter to Users**

Brief Description

This allows the admin to send mails to the registered to alert users about upcoming courses or events.

Initial Step-By-Step Description

Before this use case can be initiated, the admin must be logged in.

1. The admin panel may have additional 'Send Newsletter to users' button.
2. By clicking on above button, system will lead to another page where admin have to specify the content of Newsletter and send it by clicking on 'Send' button.

❖ Use case: **Set Training Source**

Brief Description

This allows the admin to specify the sources from which the training articles need to be collected. The admin can also specify various interesting projects & ideas for users.

Initial Step-By-Step Description

Before this use case can be initiated, the admin should enter into the admin panel.

1. The admin enters the admin panel after successful authentication.
2. The admin choses the option to set article sources.
3. The admin add / change / remove an article source.
4. The system makes the required changes.

3.0. Requirements Specification

3.1. Functional Requirements

Register

Use Case Element	Description
Use Case Description	The user requests to create an account
Primary Actor	User
Extension Points	Student , Expert
Precondition	None
Trigger	User input
Basic Flow	1. User clicks on 'Signup' button. 2. The system loads another page where user have to fill details such as Username, Email ID, and Password etc. 3. The user click on "Register" button 4. The system stores all the details in server database. Note: password should be hashed before stored to database.
Alternate Flow	1. User clicks on "Signup" and clicks on "Register" button after all the details has been input by user. 2. The system not able to store data server database due to network failure or missing user details. (a) Store user request. (b) Inform user that service could not be attended.

Search

Use Case Element	Description
Use Case Description	The user requests a description of topic
Primary Actor	User/Admin
Extension Points	Select Category(training/ideas/project)
Precondition	None
Trigger	User input
Basic Flow	5. User enters topic. 6. The system queries for available opinions on local database. 7. The system generates the list of related topics. 8. The system replies to the user.
Alternate Flow	3. User enter topic and the category is selected. 4. The system does not find opinions on server database. (a) Store user request. (b) Inform user that service could not be attended.

Submit an Idea

Use Case Element	Description
Use Case Description	The user submits their ideas
Primary Actor	User/Administrator
Precondition	User has been successfully authenticated
Trigger	User inputs data
Basic Flow	1. User/Admin inputs the description of idea in textbox. 2. The system store them persistently
Alternate flow	1. User/Admin inputs the description of idea in textbox. 2. The system does not save the contents on server database. a. Store user request. b. Inform user that service could not be attended.

3.2 Non-Functional Requirements

This Hack4innovation system will be hosted on a server with high speed Internet capability. The physical machine (PCs/Laptops/mobile devices) to be used will be determined by the user. The web application can be accessed via browsers by visiting web address or the server's IP address. The speed of the User's connection will depend on the hardware used rather than characteristics of this system.

The Admin Console will run on the server machine and is accessible to the admins by logging into the admin panel.

The functional requirements represent the main goals that the system must achieve. The non-functional requirements represent requirements that should work to assist the application to accomplish its goal.

- **Interface**

The system must provide a great user-friendly environment & attractive enough to capture the attention of normal students.

- **Performance**

The system must be able to handle a user request in an acceptable amount of time. Whenever the system cannot afford to meet the user's request, it shall schedule the task to occur in parallel (asynchronously), without blocking the flow between user and system.

- **Scalability**

The system must provide ways to retrieve as many as necessary opinions from the sources.

- **Fault-Tolerance**

The system must be able to cope with bad formatted html, network failure and parsing errors. The system shall schedule jobs automatically in face of failures.

- **Compatibility**

The system shall be compatible with any devices that in which we can install a web browser.

- **Interoperability**

Other applications can request the services of the system automatically (without direct human interaction with the GUI). The system must be able to interact with other applications using standard technologies.

- **Security**

The system must be highly secure (always ensure sanitized inputs). Since this system train students about ethical hacking there are chances some students may try these attacks on our own system. Thus this mini project sets up an example among students to build secure projects.