Indian Institute of Information Technology, Allahabad
Software Engineering

Instructors: Dr. Sonali Agarwal

SOFTWARE REQUIREMENT SPECIFICATION

FOR

Impedia

Group Members:

Shreyas Gupta	IIT2019102	iit2019102@iiita.ac.in
Harshdeep Singh Pruthi	IIT2019105	iit2019105@iiita.ac.in
Sarthak Maheshwari	IIT2019117	iit2019117@iiita.ac.in
Garvit Chittora	IIT2019142	iit2019142@iiita.ac.in

Index

Section 1: Introduction

"Impedia" is a centralized appeal management system or in more detail, it is a web application which will enable proper communication of issues or appeals made by students to appropriate faculty members or authorities for any organization, or in our use case, for IIITA.

Purpose

Currently, communication of appeals or issues within IIITA is trivial and not efficient. The current issues faced:

From Students' point of view:

- Only method of communication is through Email or Call
- Most students aren't aware of the proper authority credentials for different types of appeals thus emails or calls aren't possible until this info is known.
- Many faculties don't listen to individual emails, students have to contact their Gymkhana representatives which in turn talk to the Faculty.
- Sometimes students have to resort to mass mailing and spamming to get any decent response from authorities on a common issue.

From Faculties'/Authorities' point of view:

- Emails can be missed.
- Flooding of inboxes with appeals can cause them to miss important work related mails or important issues.

So this product aims to solve all these problems by introducing a separate platform for appeal management.

Scope

Here we will describe what features are in the scope of this product and what features are not in the scope.

In Scope:

- Flexible structure so any organization can host their separate version and their admin can add their respective authorities, faculty groups and specify their email domain so students under that domain can join the platform.

- Suggest appropriate authorities or authority groups to students based on their type of appeal and their semester/section.
- Let students make an appeal or send a message to the appropriate authority within the platform.
- Let authorities reply and resolve the appeal within the platform.
- Notify authorities or students about unread appeals/replies through email if enabled.
- Let students create common issues/appeals in the form of petitions and tag concerned authorities, which other students can view and sign, to support the issue/appeal.
- Let concerned authorities view the petition and reply to all the supporting students accordingly.

Out of Scope:

- Conversations initiated by faculties towards students.
- Creation of student groups.

Definitions, Acronyms, and Abbreviations

Abbreviations:

a. SRS: Software Requirement Specification

Definitions:

a. Authorities: Any faculty member, or any person holding any responsibility or taking decisions in the working of IIITA.
b. Admin: Any person in charge of handling and overseeing the appeal management system.

References

IEEE SRS Format

Overview

The rest of this SRS is organized as follows: Section 2 gives an overall description of the product and its features. It lists any constraints, different types of users, any assumptions or dependencies. Section 3 gives specific requirements that the product is expected to deliver. Functional requirements are given by various use cases. Some performance requirements and design constraints are also given.

Section 2: Overall Description

Product Perspective

“Impedia” is aimed to enable proper communication of issues or appeals made by students to appropriate faculty members or authorities for any organization, or in our use case, for IIITA. It will increase the efficiency of the current appeal mechanism(email) and prevent flooding of inboxes. Common issues would also get a platform instead of the usual mass mailing tactic carried out by students.

Product Functions

S. No.	Functions	Description
	Admin:	
F1.	Admin Login	Allows Admin to login.
F2.	Add Authorities	Admin can add Authorities
F3.	Set Email Domain allowed for Students	Admin can set the email domain under which the student emails are present. In case of IIITA, the domain will be iiita.ac.in
F4.	View Appeals/Petitions	Admin can view all appeals and petitions on the platform and monitor them.
F5.	Make Authority Group	Admin can make the groups of authorities for departments or courses.
F6.	Add authorities to group	Admin can add appropriate authorities to the respective group.
	Authorities:	
F7.	Authority Login	Authorities can log in on the platform through credentials created by Admin.
F8.	View Appeal	Authorities can view the appeal

		created by the student.
F9.	Reply to appeal	Authorities can reply to the appeals made by the students.
F10.	Receive Notifications	Authorities will get notifications of any appeals through their email.
F11.	Change Password	Authorities can change their password.
F12.	Update Profile	Authorities can update their profile.
F13.	View Petition	Authorities can view the petition created by some other student.
F14.	Decision on the petition	Authorities can announce their decision based on the number of signs on the petition.
	Students:	
F15.	Student Login	Students can log in on the platform
F16.	Make appeal	Students can make an appeal to a particular authority or a group.
F17.	View Reply	Students can view the reply of the Authorities on their appeal.
F18.	Change Password	Students can change their password.
F19.	Update Profile	Students can update their profile.
F20.	Create Petition	Students can create a petition of the issue they are facing.
F21.	View Petition	Students can view any petition created by them or by any other student.
F22.	Sign Petition	Students can sign a petition to support the issue.

User Classes and Principal Actors

Three types of people will use the web application: the admins, the authorities, and the students

Admins: Admins will be responsible for adding authorities and creating appropriate authority groups for different types of departments. Admins will also set the email domain under which the students have email accounts, for IIITA, the domain will be iiita.ac.in. Admins will also have the ability to view all messages and appeals being exchanged to keep a check.

Authorities: Authorities will be able to view all individual appeals made by students or any common petitions created by a student group and will be able to reply to them accordingly. They can also allow email notifications for any new appeals they receive, for speeding up the process.

Students: Students will be able to make appeals/send messages/create petitions for their issues or concerns. They can enter their appeal type and the platform will suggest the correct authorities to make the appeal to. Students can also view petitions created by others and can choose to support them. Students can view replies or resolutions received for any appeal made by them. They can also allow email notifications for any new replies they receive, for speeding up the process.

General Constraints

a. Using the Web Application requires Internet connectivity.

Assumptions and Dependencies

a. Working of the Web Application requires Internet connectivity.
b. The Admin must add all the authorities, create authority groups and set the student email domain before the Web Application will be usable for users to make appeals.

Section 3: Specific Requirements

Functional Requirements

Here we will describe the functional requirements by mentioning all the use cases.

Use Case 1:

Name: Admin login
Description: Allows Admin to login.
Actors: Admin
Pre-conditions:

- Internet connectivity.
- Only for allowed mail ids.

Main success scenario:

- Admin enters credentials and clicks on the login button.
- App checks for the validity of the credentials.

Extension:
Id or Password incorrect. Error message is displayed.
Post-condition:
Admin can now access all features of the app.

Use Case 2:

Name: Add Authorities
Description: Allows Admin to add authorities/faculty to access the application.
Actors: Admin
Pre-conditions:

- Internet connectivity.
- Only for allowed mail ids.

Main success scenario:

- Admin enters credentials of authorities to add and clicks on the invite button.
- App sends an invitation on provided Email ID to join the platform in the respective user category.

Extension:
Email ID entered was incorrect. This particular person did not get access to the platform.

Post-condition:
Authorities can now access the respective features of the app.

Use Case 3:

Name: Set Email Domain allowed for Students
Description: Admin can set the email domain under which the student emails are present
Actors: Admin
Pre-conditions:

- Internet connectivity.

Main success scenario:

- Admin enters the email domain under which the college's students have email accounts.

Extension:
Domain was invalid. Error is displayed.
Post-condition:
Students with emails under the entered domain and now come and join the platform.

Use Case 4:

Name: View Messages
Description: Allows Admin to view all messages/conversations.
Actors: Admin
Pre-conditions:

- Internet connectivity.
- Only for allowed mail ids.

Main success scenario:

- Admin chooses a particular appeal and clicks on it.
- App shows the appeal information - title, description group that raised it and the conversation that is happening between authorities and students.

Extension:
NIL
Post-condition:
Admin can now view all the messages of a particular appeal.

Use Case 5:

Name: Make Authority Group
Description: Allows admin to make groups of authorities for departments or courses.
Actors: Admin
Pre-conditions:

- Internet connectivity.
- Only for allowed mail ids.

Main success scenario:

- Admin enters mails IDs of authorities and creates their group respectively.
- App checks for the validity of the Mail IDs.

Extension:
Mail ID was incorrect. This particular person is not added to the group.
Post-condition:
New group is created which is accessible by respective group members.

Use Case 6:

Name: Add authorities to group
Description: Allows admin to add appropriate authorities to the respective group.
Actors: Admin
Pre-conditions:

- Internet connectivity.
- Only for allowed mail ids.

Main success scenario:

- Admin enters the Mail ID of authorities and in the Add Authority section of a group and clicks on Add Authority.
- App checks for the validity of the Mail ID.

Extension:
Mail Id incorrect. The new user was not added.
Post-condition:
The particular user also belongs to the group now.

Use Case 7:

Name: Authorities Login
Description: Allows authorities to login to the platform through the credentials created by Admin.

Actors: Authorities
Pre-conditions:

- Internet connectivity.
- Only for allowed mail ids.

Main success scenario:

- A person in the Authority enters the credentials and clicks on the Login button.
- App checks for the validity of the credentials.

Extension:
Mail ID or password incorrect. Error message is displayed.
Post-condition:
The particular user from the authority is now logged in.

Use Case 8:

Name: View Appeal
Description: Allows authorities to view any appeal (concerned with their group) raised by the students
Actors: Authorities
Pre-conditions:

- Internet connectivity.
- Only for allowed mail ids.

Main success scenario:

- A person in the Authority selects any Appeal from the Appeals Section and clicks on View Appeal.
- App makes sure that the appeal to be viewed falls in the user's group.

Extension:
Appeal is not meant for any user of this group. The user is rerouted to their group's Appeal Page.
Post-condition:
The user from the authority can now see all the information regarding the selected Appeal - Raised by, Title, Description and the messages involved in it.

Use Case 9:

Name: Reply to Appeal
Description: Allows authorities to reply to any concerned Appeal.
Actors: Authorities

Pre-conditions:

- Internet connectivity.
- Only for allowed mail ids.

Main success scenario:

- A person in the Authority selects an Appeal, enters the message in the Chat Box and clicks on the Send button.

Extension:

NIL

Post-condition:

The message was sent and now is visible to the student who raised it.

Use Case 10:

Name: Receive Notifications

Description: Allows authorities to receive Notifications of any recent chats through Email.

Actors: Authorities

Pre-conditions:

- Internet connectivity.
- Only for allowed mail ids.

Main success scenario:

- A person in the Authority can select the Receive Notifications option in the Settings section.

Extension:

NIL

Post-condition:

The user will now receive notifications of any recent chats through the mail.

Use Case 11:

Name: Change Password

Description: Allows authorities to change their password.

Actors: Authorities

Pre-conditions:

- Internet connectivity.
- Only for allowed mail ids.

Main success scenario:

- User clicks on the Change Password in Profile section.

- Enters old password, new password and confirms new password.
- App checks for the validity of the credentials.

Extension:
Old Password incorrect. Password not changed.
Post-condition:
The user has successfully changed their password.

Use Case 12:

Name: Update Profile
Description: Allows Authorities' users to update their Profile.
Actors: Authorities
Pre-conditions:

- Internet connectivity.
- Only for allowed mail ids.

Main success scenario:

- User clicks on Update Profile in Profile section.
- Prompted by an Update Dialog, user changes the necessary information - like Name, Picture, etc.

Extension:
NIL
Post-condition:
The user has updated their profile.

Use Case 13:

Name: View Petition
Description: Allows authorities to view petitions.
Actors: Authorities
Pre-conditions:

- Internet connectivity.
- Only for allowed mail ids.

Main success scenario:

- User selects a particular petition from the Petitions section and clicks on View Petition.

Extension:
NIL
Post-condition:

The user can now see the petition - its description, students who have signed it and who created the petition.

Use Case 14:

Name: Decision on the Petition.
Description: Allows authorities to post a decision on the raised petition.
Actors: Authorities
Pre-conditions:

- Internet connectivity.
- Only for allowed mail ids.

Main success scenario:

- User selects a particular petition from the Petitions section and can post a final decision on it.

Extension:
NIL
Post-condition:
The decision has now been posted and is visible to the students.

Use Case 15:

Name: Student Login.
Description: Allows students to login to the platform through the credentials created by Admin.
Actors: Students
Pre-conditions:

- Internet connectivity.
- Only for allowed mail ids.

Main success scenario:

- Student enters the credentials and clicks on the Login button.
- App checks for the validity of the credentials.

Extension:
Mail ID or password incorrect. Error message is displayed.
Post-condition:
The particular user from the authority is now logged in.

Use Case 16:

Name: Make Appeal

Description: Allows any student to make any appeal to the authorities.
Actors: Students
Pre-conditions:

- Internet connectivity.
- Only for allowed mail ids.

Main success scenario:

- Student clicks on New Appeal in the Appeals section.
- Provides details on the appeal - title, description and selects the valid authority group to send the appeal to

Extension:
NIL
Post-condition:
A new appeal to the selected authority group is now submitted.

Use Case 17:

Name: View Reply
Description: Allows students to see a reply by authorities to any Appeal.
Actors: Students
Pre-conditions:

- Internet connectivity.
- Only for allowed mail ids.

Main success scenario:

- Student selects an Appeal he/she created in the Appeals section.

Extension:
NIL
Post-condition:
The appeal shows up with all the replies by the authorities, if any.

Use Case 18:

Name: Change Password
Description: Allows students to change their password.
Actors: Students
Pre-conditions:

- Internet connectivity.
- Only for allowed mail ids.

Main success scenario:

- User clicks on the Change Password in Profile section.
- Enters old password, new password and confirms new password.
- App checks for the validity of the credentials.

Extension:
Old Password incorrect. Password not changed.
Post-condition:
The user has successfully changed their password.

Use Case 19:

Name: Update Profile
Description: Allows Students to update their Profile.
Actors: Students
Pre-conditions:

- Internet connectivity.
- Only for allowed mail ids.

Main success scenario:

- User clicks on Update Profile in Profile section.
- Prompted by an Update Dialog, the user changes the necessary information - like Name, Picture, etc.

Extension:
NIL
Post-condition:
The user has updated their profile.

Use Case 20:

Name: Create Petition
Description: Allows any student to create a petition for any issue they are facing.
Actors: Students
Pre-conditions:

- Internet connectivity.
- Only for allowed mail ids.

Main success scenario:

- Student clicks on New Petition in the Appeals section.
- Provides details on the petition - title, description and selects the valid authority group (default - all groups) to send the appeal to.

Extension:

NIL

Post-condition:

A new petition is created visible to concerned authorities and all students which they can support by signing for it.

Use Case 21:

Name: View Petition

Description: Allows Students to view petitions.

Actors: Students

Pre-conditions:

- Internet connectivity.
- Only for allowed mail ids.

Main success scenario:

- Student selects a particular petition from the Petitions section and clicks on View Petition.

Extension:

NIL

Post-condition:

The user can now see the petition - its description, students who have signed it and who created the petition.

Use Case 22:

Name: Sign Petition

Description: Allows Students to sign any petitions.

Actors: Students

Pre-conditions:

- Internet connectivity.
- Only for allowed mail ids.

Main success scenario:

- Student selects a particular petition from the Petitions section and clicks on Sign Petition.

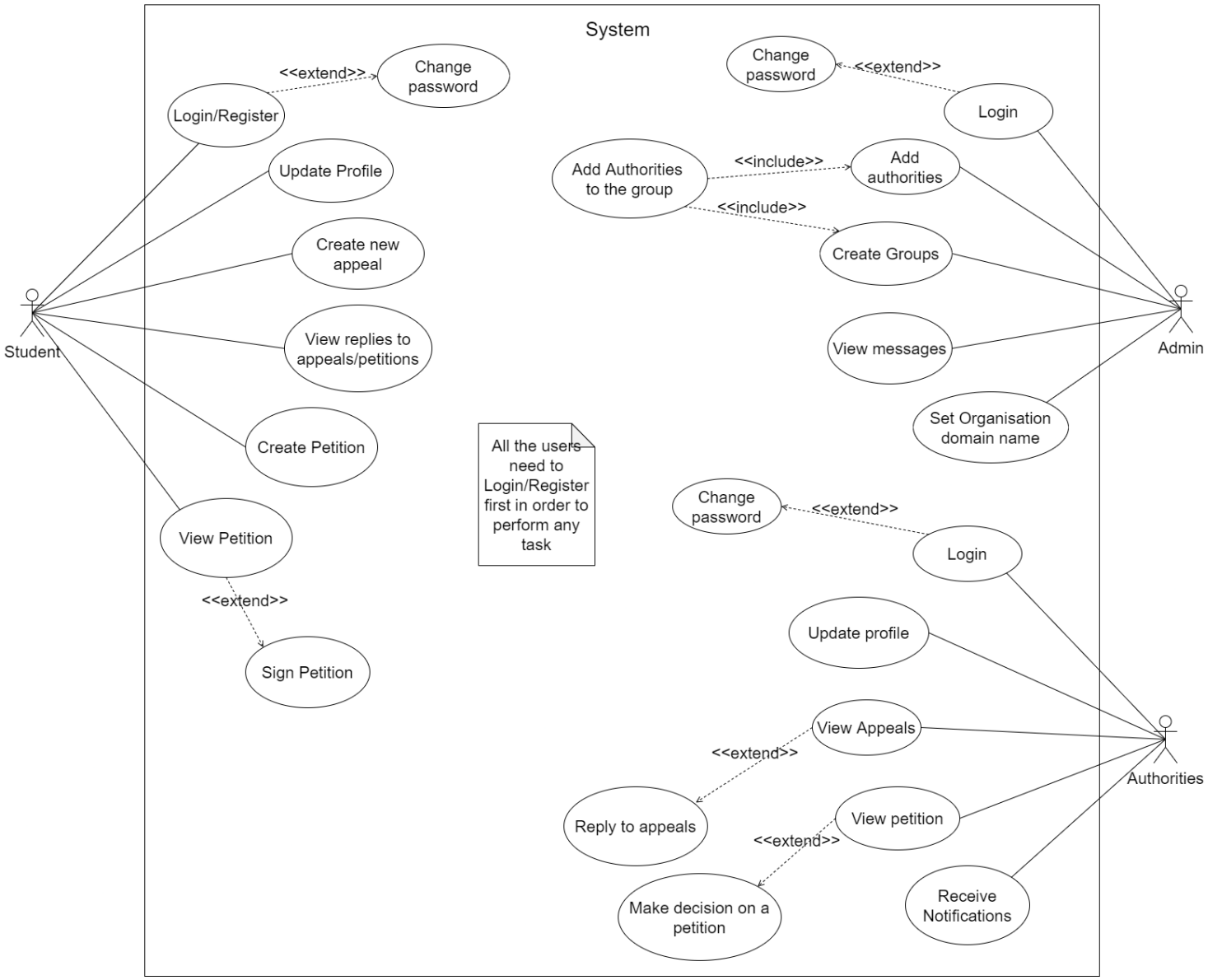
Extension:

NIL

Post-condition:

The user can has now signed the petition and his/her name is visible in the list of signees.

Use Case Diagram



Use Case Diagram

Non-functional Requirements

1. **Product Requirements:**
    - The app must load between 2-4 seconds, to minimize the bounce rates
    - HTTPS Protocol must be enforced on all pages to ensure secure data transfer.
    - Data should be secured in an event of system failure.
    - User inputs must be sanitised to prevent potential XSS attacks.
    - The system must automatically log a user out after a period of inactivity.
    - The backup of the database is continuously maintained and updated to reflect the most recent changes.
    - The application should be compatible with different environments so that the user can access it without any major inconvenience.
    - The application should be Search Engine Optimized to increase the page rank of the application in search results.

2. **Organisational Requirements:**
    - Users must login with their organisation E-mail address to access the portal.
    - The application should follow legal and compliance rules.

3. **External Requirements:**
    - A sufficiently modern browser to access all the features of the app without causing any compatibility issues.