

**Indian Institute of Information Technology,
Allahabad**

Software Engineering

SOFTWARE REQUIREMENT SPECIFICATION

AMIGO AMS APPLICATION

GROUP MEMBERS:

IIT2019194 (RAHUL ROY)

IIT2019230 (ESHAN VAID)

IIT2019198 (NISCHAY NAGAR)

IIT2019240 (AYUSH KHANDELWAL)

Table of Contents

Table of Contents	ii
1. Introduction	1
1.1 Purpose	1
1.2 Document Conventions	1
1.3 Intended Audience and Reading Suggestions	1
1.4 Product Scope	1
1.5 References	1
2. Overall Description	2
2.1 Product Perspective	2
2.2 Product Functions	2
2.3 User Classes and Characteristics	2
2.4 Design and Implementation Constraints	2
2.5 Assumptions and Dependencies	3
3. External Interface Requirements	3
3.1 User Interfaces	3
3.2 Hardware Interfaces	3
3.3 Software Interfaces	3
3.4 Communications Interfaces	3
4. System Features	4
4.1 System Feature 1	4
4.2 System Feature 2 (and so on)	4
5. Other Nonfunctional Requirements	4
5.1 Performance Requirements	4
5.2 Security Requirements	5
5.3 Software Quality Attributes	5
Appendix A: Glossary	5
Appendix B: Analysis Models	5

1. Introduction

“Amigo” is basically a web application that offers a user-friendly interface for management of academic affairs (Admissions, Assessments, Awards and Archives).

1.1 Purpose

The process of admission and academic management of our institution is less efficient and clumsy in our organisation (NU), as per one of the spokesperson from the Academic Department. Hence we aim to create a platform for smooth, easy and efficient interaction between the students, academic administration and FnA.

From student's point of view:

- . Student's find it difficult to complete the registration and admission process as on the day of admission, many students arrive at the spot at the same time which makes it a difficult, slow and less efficient procedure.
- . Students want a platform where they can know the admission related procedure in advance so that it does not get complicated on the day of registration.
- . All institution holidays, institute calendar and course details must be provided in a simple and structured way to students.

From Academic staff's point of view:

- . The academic management gets a heavy load of work on the day of admission, due to simultaneous registration processes taking place.
- . They want a platform to provide information regarding the institution calendar, events and more to all the students without worrying about any future updates, reschedulement and cancelation.

So, this project is an attempt to overcome these drawbacks in IIIT Allahabad.

1.2 Document Conventions

- 1) *Text highlighted in the srs means a special significance or importance to the reader or any institutional staff of IIIT Allahabad.*
- 2) *Following are the abbreviations used:*
 - AM - Academic Management*
 - AMS - Academic Management System*
 - RPS - Registration Procedure System*
 - “Amigo©”:: Copyrighted app name.*
 - SRS: Software Requirement Specification*

- 3) "Amigo" is an independent group project which does not share any similarity with other app.

1.3 Intended Audience and Reading Suggestions

This document is meant for the following readers:

- .) Amigo App Developer*
- .) Academic staff of IIIT Allahabad*
- .) Project Evaluator*

The SRS starts with the table of contents where the reader may find the content to look forward to that interests him/her. Then we introduce our aim and the purpose of making the project, After that the reader may find the use case diagram, brief description of the use cases, the non-functional and functional requirements and finally the design model of the app.

1.4 Product Scope

We describe what features are in the scope of the software and what are not in the scope of the software to be developed for the AMS..

In Scope:

FOR HoD/Coordinator:

1. Profile view (Personnel details) and request for modifications if any
2. Academic calendar display
3. Faculty list management
4. Creation/updation of courses (Core, Elective)
5. Batch-wise, program-wise, semester-wise allotment of courses
 - a. Core
 - b. Elective portal management
 - c. Project portal management (Summer internship/master projects)
6. Batch-wise, program-wise, semester-wise course registration status
7. Class time-table upload and display
8. Exam schedule upload and display
9. Appointment/update of course coordinator
10. Notice board management
11. General messages / notifications based on event

FOR FnA:

1. Fees chart management
2. Fees payment status at the time of admission
3. Fees payment status at the time of registration
4. Fees approve/disapprove

5. Fees waiver status
6. Defaulters list display
7. Special permission granted for delayed payment of fees
8. Reconciliation of fees

Out of Scope:

- a. Verification of original marksheet of 12th / any engineering entrance exam i.e, JEE is difficult and in efficient
- b. There is no communication among the IIIT Allahabad and Industries via this application.

1.5 References

- A. IEEE Software Engineering Standards Committee, "IEEE Std. 830-1998, IEEE Recommended Practice for Software Requirements Specifications, October 20, 1998.
- B. <https://www.iiita.ac.in/academics/department/>

2. Overall Description

The rest of this SRS is organized as follows: Section 2 gives an overall description of the software. It gives what level of proficiency is expected of the user, some general constraints while making the software and some assumptions and dependencies that are assumed. Section 3 gives specific requirements which the software is expected to deliver. Functional requirements are given by various use cases. Some performance requirements and design constraints are also given.

2.1 Product Perspective

"Amigo©" is aimed to increase the pace and simplicity of academic affairs of IIIT Allahabad and effective conduct of the Admission cum Registration process and to overcome the problems of current management faced by both, Students as well as the academic staff of IIIT Allahabad. One-click feature of the apps will reduce the paperwork, save a lot of time for students who have to visit the Registration Cell for small reasons at times. Overall, this application will reduce the trouble of academic management staff, of records management of students, communication with them in case of an emergency, etc.

2.2 Product Functions

"Amigo©" supports the following use cases:

Serial No.	Use Cases	Description of Use Cases
<i>HOD/Coordinator :</i>		
1	<i>Authorized Login and Profile View</i>	<i>allows user to check profile and request modification</i>
2	<i>Calendar Display</i>	<i>allows user to display the academic calendar from database</i>
3	<i>Faculty List Management</i>	<i>allows user to manage the list of faculties i.e, insert, delete or modify the faculty info</i>
4	<i>Course Modification</i>	<i>allows user to create and update courses</i>
5	<i>Course Allotment</i>	<i>allows user to allot courses batch-wise, semester-wise and program wise</i>
6	<i>Course Registration Status</i>	<i>allows the user to check the status of course registration for various batches or programs</i>
7	<i>Class TimeTable Display</i>	<i>allows user to upload and display class time table for all batches and programs</i>
8	<i>Exam Schedule Display</i>	<i>allows user to upload and display exam schedule for all batches and programs</i>
9	<i>Appoint Course Coordinator</i>	<i>allows user to a coordinator to a particular course</i>
10	<i>Manage Notice Board</i>	<i>allows user to manage Notice Board</i>
11	<i>Notification</i>	<i>allows user to notify on future events and schedules</i>
<i>FnA:</i>		
1	<i>Authorized Login</i>	<i>allows user to login</i>
2	<i>Manage Fees Chart</i>	<i>allows user to manage the fees chart of all students</i>
3	<i>Check Fee Status</i>	<i>allows user to check the fees status at the time of registration and admission</i>
4	<i>Fee Approval</i>	<i>allows user to approve/disapprove the fees payment by any student</i>
5	<i>Fee Waiver Status</i>	<i>allows user to check and update the fees waiver</i>
6	<i>Defaulters List Display</i>	<i>allows user to display the list of all the</i>

		<i>defaulters who have not paid the fees</i>
7	<i>Late Fee Permission</i>	<i>allows user to provide late fee fine, extended deadline and permission for delayed fee</i>
8	<i>Fee Verification</i>	<i>allows user to verify fees payment of all valid students who have enrolled in various programs</i>

2.3 User Characteristics

The user should be familiar with the operation of Android Device and Web Applications.

2.4 Principal Actors

“Amigo©” is going to have 3 principal actors namely:

- 1. System Administrator*
- 2. HoD/ Co-ordinator*
- 3. Finance Administrator*

2.5 Design and Implementation Constraints

- 1. Working on “Amigo©” requires Internet connection.*
- 2. “Amigo©”; is a single-user Application. Every student must have this application on their android device.*
- 3. All the faculty members and academic staff including the members of the FnA must be registered on this platform.*

2.6 Assumptions and Dependencies

- a. Working of “Amigo©” is dependent on the availability of Internet connection.*
- b. User Authentication is dependent on the LDAP server.*
- c. The System Administrator must check the validity of each user and update the requests made by each user from time to time.*

3. External Interface Requirements

3.1 User Interfaces

The six interface types found in the website are as follows:

- 1. UI of the dashboard*
- 2. UI of the Profile View*
- 3. UI of the Course Selection*
- 4. UI of the Fee Portal*
- 5. UI of the Academic calendar*
- 6. UI of the Time Table and other informations*

3.2 Hardware Interfaces

All the hardware on which a Web Browser can be installed can act as an interface for the product and the user. The Hardware can be mobile, laptop, pc and tablet.

3.3 Software Interfaces

All the software required for a Web Browser needs to be installed and act as an interface for the product and the user.

3.4 Communications Interfaces

The communication between the different parts of the system is important since they depend on each other. However, in what way the communication is achieved is not important for the system and is therefore handled by the underlying operating systems for web application. But, any transaction on the web deals with the following protocols.

- TCP/IP*
- HTTP*

4. System Specific Requirements

4.1 Functional Requirements

HOD/Coordinator-

Use Case 1:

Name: Authorized Login and Profile View

Summary: Allows users to check profile and request modification.

Actors: System Admin, HOD

Pre-conditions:

- Internet connectivity.
- Only for allowed mail ids.

Main success scenario:

- Admin clicks on login button.
- App checks for the authorization of login.

Extension:

Id or password incorrect. Shows error dialog box.

Post-condition:

Admin can now access all features of the app.

Use Case 2:

Name: Calendar Display

Summary: Allows user to display the academic calendar from database

Actors: System Admin, HOD and FnA

Pre-conditions:

- Internet connectivity.
- Only for allowed mail ids.

Main success scenario:

- Admin clicks on the calendar button.
- App shows the calendar.

Extension:

NIL

Post-condition:

NIL

Use Case 3:

Name: *Faculty List Management*

Summary: *Allows user to manage the list of faculties i.e, insert, delete or modify the faculty info*

Actors: System Admin, HOD and FnA

Pre-conditions:

- Internet connectivity.
- Only for allowed mail ids.

Main success scenario:

- Admin clicks on the faculty List Management button.

Extension:

NIL

Post-condition:

NIL

Use Case 4:

Name: *Course Modification*

Summary: *allows user to create and update courses*

Actors: System Admin, HOD and FnA

Pre-conditions:

- Internet connectivity.
- Only for allowed mail ids.

Main success scenario:

- Admin clicks on the *Course Modification* button.
- Admin modifies the course

Extension:

NIL

Post-condition:

NIL

Use Case 5:

Name: *Course Allotment*

Summary: *allows user to allot courses batch-wise, semester-wise and program wise*

Actors: System Admin, HOD and FnA

Pre-conditions:

- Internet connectivity.
- Only for allowed mail ids.

Main success scenario:

- Admin clicks on the Course *Allotment* button.

Extension:

NIL

Post-condition:

NIL

Use Case 6:

Name: *Course Registration Status*

Summary: *allows the user to check the status of course registration for various batches or programs*

Actors: System Admin, HOD and FnA

Pre-conditions:

- Internet connectivity.
- Only for allowed mail ids.

Main success scenario:

- Admin clicks on the Course *Registration Status* button.

Extension:

NIL

Post-condition:

NIL

Use Case 7:

Name: *Class TimeTable Display*

Summary: *allows user to upload and display class time table for all batches and programs*

Actors: System Admin, HOD and FnA

Pre-conditions:

- Internet connectivity.

- Only for allowed mail ids.

Main success scenario:

- Admin clicks on the Class *TimeTable Display* button.

Extension:

NIL

Post-condition:

NIL

Use Case 8:

Name: *Exam Schedule Display*

Summary: *allows user to upload and display exam schedule for all batches and programs*

Actors: System Admin, HOD and FnA

Pre-conditions:

- Internet connectivity.
- Only for allowed mail ids.

Main success scenario:

- Admin clicks on the Exam *Schedule Display* button.

Extension:

NIL

Post-condition:

NIL

Use Case 9:

Name: *Appoint Course Coordinator*

Summary: *allows user to a coordinator to a particular course*

Actors: System Admin, HOD and FnA

Pre-conditions:

- Internet connectivity.
- Only for allowed mail ids.

Main success scenario:

- Admin clicks on the Appoint *Course Coordinator* button.

Extension:

NIL

Post-condition:

NIL

Use Case 10:

Name: *Manage Notice Board*

Summary: *allows user to manage Notice Board*

Actors: System Admin, HOD and FnA

Pre-conditions:

- Internet connectivity.
- Only for allowed mail ids.

Main success scenario:

- Admin clicks on the *Manage Notice Board* button.

Extension:

NIL

Post-condition:

NIL

Use Case 11:

Name: *Notification*

Summary: *allows user to notify on future events and schedules*

Actors: System Admin, HOD and FnA

Pre-conditions:

- Internet connectivity.
- Only for allowed mail ids.

Main success scenario:

- Admin clicks on the *Notification* button.

Extension:

NIL

Post-condition:

NIL

FnA-

Use Case 12:

Name: Authorized Login

Summary: Allows user to login as a FnA .

Actors: System Admin, FnA

Pre-conditions:

- Internet connectivity.
- Only for allowed mail ids.

Main success scenario:

- Admin clicks on login button.
- App checks for the authorization of login.

Extension:

Id or password incorrect. Shows error dialog box.

Post-condition:

Admin can now access all features of the app.

Use Case 13:

Name: Manage Fees Chart

Summary: Allows user to manage the fees chart of all students (i.e, date and time of payment, late fee extension and permission, amount paid, e.tc)

Actors: System Admin, FnA

Pre-conditions:

- Internet connectivity.
- Only for allowed mail ids.

Main success scenario:

- Admin clicks on “Manage Fees Chart” button.
- App lists the details of the fees in a chart.

Extension:

NIL

Post-condition:

Admin can now access all details of the fees of all students in an organized manner

Use Case 14:

Name: Check Fee status

Summary: Allows user to check the fee payment and receive status. User can also know the details regarding Fee waiver.

Actors: System Admin, FnA

Pre-conditions:

- Internet connectivity.
- Only for allowed mail ids.
- > password required at check

Main success scenario:

- Admin clicks on “Check Fee Status” button.
- App checks for password and delivers the required information regarding fees.

Extension:

Admin can notify students directly about their fee status.

Post-condition:

Admin can now check the fee status.

Use Case 15:

Name: Fee Approval

Summary: Allows users to give permission for late fee and verify fee deposition by student..

Actors: System Admin, FnA

Pre-conditions:

- Internet connectivity.
- Only for allowed mail ids.

Main success scenario:

- Admin clicks on “Approve Fee” button.
- App checks for the authorization of login and sets the approval status to true.

Extension:

Admin can notify students when they get approval for late fee.

Post-condition:

Admin can now approve the fee deposition by students.

Use Case 16:

Name: Defaulters List Display

Summary: Allows user to see the list of all defaulters who have not submitted the fees

Actors: System Admin, FnA

Pre-conditions:

- Internet connectivity.
- Only for allowed mail ids.

Main success scenario:

- Admin clicks on the “Display Defaulters List” button.
- App lists out the defaulters in alphabetical order / by order of their class and roll number

Extension:

Admin can give a warning message directly to all students at once who have not submitted the fees.

Post-condition:

Admin can now check the list of defaulters

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The app should be suitable for all device screen sizes.

The class-timetable and academic calendar should be updated on a regular basis efficiently.

5.2 Security Requirements

Information regarding the institute, its events and all related affairs that is important for the dignity and safety of the institute and its surroundings should be made private and kept with strongly encrypted codes. Information regarding students and fee payment should not be made public and only accessible by the FnA or HoD.

5.3 Software Quality Attributes

The user interface (UI) should support both modern and old forms of styles that can help users work according to their styles and usage.

Appendix A: Glossary

1. HTTP: Hypertext Transfer Protocol.
2. HTML: Hyper-text markup language
3. CSS: Cascading style sheets
4. JS: Java Script

Appendix B: USE-CASE DIAGRAM

