
Software Requirements Specification

for

NUApp

Version 1.0 approved

Prepared by NVS Abhilash

NIIT UNIVERSITY

18 September 2016

Table of Contents

Table of Contents	ii
Revision History	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 Operating Environment	2
2.5 Design and Implementation Constraints	2
2.6 User Documentation	2
2.7 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 Safety Requirements	5
5.3 Security Requirements	5
5.4 Software Quality Attributes	5
5.5 Business Rules	5
6. Other Requirements	5
Appendix A: Glossary	5
Appendix B: Analysis Models	5
Appendix C: To Be Determined List	6

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The product is a mobile application to help students of NIIT University to be able to view their current timetable with ease; to be able to apply gatepass through a mobile application itself and the application must notify the student and concerned parent / guardian about the same; to be able to view their issued books, to be notified about the reissue date; to be able to search for specific book availability in the library; and to view their “non-detail” attendance in the app.

1.2 Document Conventions

*Font family: Arial
Font size: 11
Headings are bold*

1.3 Intended Audience and Reading Suggestions

This document is mainly intended for anyone who is interested to know more about the project, any faculty, student or developer is welcomed to read this document and if possible please send us your feedback if improvements is needed.

1.4 Product Scope

Purpose:

Software being specified is X, an hybrid mobile application which is currently just intended for students of NIIT University, to enable them to check their timetable; apply gatepass (a kind of approval from their wardens to go outside NIIT University campus); query availability of books in the library, check their issued books return deadline and see their recent attendance.

Benefits:

- The application shall help those students who would always have to manually go to the ERP login and then check their timetable. This application shall notify the student of the upcoming class directly from the student's phone.*
- The application shall be useful to those students who would have to check their recent attendance all the time, the app provides an easy “over simplified” version of the attendance.*
- The application shall help those students who tend to forget to apply gatepass in the last moment, and face difficulty due to the long login procedure of the moodle. Instead the application provides an easy 3-5 step user interface to do the same job in a smooth and simple way.*

- The application shall send a notification of the check-in or check-out activity of the student to their parents, and student itself, which is a great safety precaution.
- The application shall help those who had to go all the way to the Library to issue a book but end up knowing that book is not available. The application shall be able to search for the availability of the book in the library.
- The application shall provide a way to notify users about the reaching deadline of the return of the books in the library.
- The application is an hybrid application which enables the application to target all the popular platforms like Android, iOS and Windows.

What the Application is not:

- The application shall not be a replacement of the attendance module in the ERP, the application would just show a simplified way of the recent attendance, i.e. of just the recent term. To view the comprehensive details student must use the ERP for the same.
- Application is also not an replacement of the Gatepass system of the University. It just would be able to request local and outstation gatepass, to experience the full functionality of the gatepass, one must use the assigned website for the gatepass.
- Application is not a way to issue or reissue the book. The student must be physically present in the Library for the same.

1.5 References

The Application include features of the following websites:

- <https://moodle.niituniversity.in/moodle/>
- <https://nucleus.niituniversity.in/>

2. Overall Description

2.1 Product Perspective

This product is a complementary application to the services provided by the University like Gatepass, Library, timetable and attendance. The product cannot replace this services, but would act as a great benefit for the students.

2.2 Product Functions

Functions of the application:

- User must login with their University e-mail account in order to experience the app features.
- App shall notify user of every check-in and check-out from the university.
- App shall notify user of the return deadline of the issued library books.
- App shall be able to notify next class timings and location to the logged in student.
- User shall be able to apply Local and Outstation gatepass from the app.
- User can check the availability of the books in the library.
- App shall be able to notify user of the issue details to the student.

- Student shall be able to check his/her recent attendance.

2.3 User Classes and Characteristics

The App has a single user class, NU Students who will use the product functions defined above for the ease of using the current system. The students are the potential customers. The App aims to simplify the above mentioned three systems and allow the students to access them with ease and reliability.

2.4 Operating Environment

The application development shall be performed under the following environment:

- Operating System: Ubuntu 16.04 (not a must but is used by the team for resonance in installation of software, and frameworks)
- Text Editor: VSCode and Atom.
- Framework: Ionic framework, includes Angular and Cordova framework.
- Programming or development language: TypeScript, HTML, CSS, SASS, JS.

2.5 Design and Implementation Constraints

- For the application to work it must interface with the present working Gatepas system.
- The application must connect with the Library Koha Database.
- The application must call Nucleus API to retrieve timetable and attendance.
- Due to security issues developers might not get complete access to these above services resulting in limiting the functionality developers can add to the application.
- Developers need to have proper experience in Web Development, in Angular framework, Ionic Framework and TypeScript to fast track the development period.

2.6 User Documentation

Till now no such documents have been prepared and will be prepared after some of the development is done.

2.7 Assumptions and Dependencies

- Developers know they might not get full access to the services resulting in limiting functionality.
- Developers as of making of this document are not sure how to access the Library Koha Database, as the Koha API is not well written. They might have to directly connect to the Koha Database, which limits the use of Library Module from University's network only.
- It is assumed that internet connection to mobile devices would be provided by the time the application is developed.

3. External Interface Requirements

3.1 User Interfaces

The user interface is kept very simple for the ease of the students and is not overloaded with features and buttons. The Buttons are made of cards. The pages are to be moved in a sliding format. The side menu bar will be available at every page. Page navigation is through sliding. Due date is displayed at the homepage itself with red color label to signify the importance.

3.2 Hardware Interfaces

The application supports mobile devices running on Android, ios and Windows

3.3 Software Interfaces

The application connects with the following interfaces:

- 1. Shall connect to the Gatepass Database through a REST API which would also be developed by the developers of the project.*
- 2. Shall connect to the Nucleus API (developed by the TCO team of NIIT University) to get Timetable and attendance from the database.*
- 3. Shall connect to the Koha Database of the Library to get details of library related operations.*
- 4. Shall connect to the Google Firebase, for authentication and notification to student about library and gatepass details.*
- 5. Shall have local storage to store the timetable and other details of the app.*

3.4 Communications Interfaces

The application for now just expect to use FCM (previously GCM) to send notification to the client side application.

4. System Features

Identifier	Short Name	Short Description
REQ1	Apply Local Gatepass	<i>This feature aims to provide students to apply local gatepass in one go. This feature helps students to remove all the</i>

Identifier	Short Name	Short Description
		<i>overhead to log in to moodle and then open gatepass module, go to local gatepass and then apply for gatepass.</i>
REQ 2	Apply Outstation Gatepass	<i>This feature helps the user to apply a outstation gatepass with ease and removes the same overhead described in REQ1</i>
REQ 3	Notifications for gatepass	<i>This feature is responsible to send notification to the student when requested gatepass is approved/ rejected, when student checks out and checks in.</i>
REQ 4	Notification for library	<i>This feature notifies user when they issued a book, and also reminds users when the return date is close.</i>
REQ 5	Query book availability in Library.	<i>This feature helps student to search for the number of copies present in library of a specific book.</i>
REQ 6	View Timetable	<i>This is a module inspired by another similar app, Intensio. This feature takes timetable input from Nucleus API once, stores in local database and then displays the data in a good intuitive UI.</i>
REQ 7	Show attendance	<i>This feature is non-functional feature,</i>

Identifier	Short Name	Short Description
		<i>which shows the attendance of the student in the recent term only.</i>

4.1 Apply Local Gatepass

4.1.1 Description and Priority

This feature aims to provide students to apply local gatepass in one go. This feature helps students to remove all the overhead to log in to moodle and then open gatepass module, go to local gatepass and then apply for gatepass.

Priority: 9

4.1.2 Stimulus/Response Sequences

Response sequence:

- 1. User selects Apply Gatepass from app.*
- 2. User selects Local Gatepass*
- 3. User gets notified for the confirmation.*
- 4. Done.*

4.1.3 Functional Requirements

REQ-1: Working UI in the mobile device

REQ-2: Internet connection

REQ-3: Student must not be black-listed (gatepass blocked)

REQ-4: University's local server must be up.

4.2 Apply Outstation Gatepass

4.1.1 Description and Priority

.

Priority: 6

4.1.2 Stimulus/Response Sequences

Response sequence:

- 1. Select Outstation gatepass*
- 2. Fill in destination and purpose .*
- 3. Apply*
- 4. Get notified when warden approves your request.*

4.1.3 Functional Requirements

REQ-1: Working UI in the mobile device

REQ-2: Internet connection

- REQ-3: Student must not be black-listed (gatepass blocked)
- REQ-4: University's local server must be up.

4.3 Notifications for Gatepass

4.1.1 Description and Priority

This feature is responsible to send notification to the student when requested gatepass is approved/ rejected, when student checks out and checks in.

Priority: 9

4.1.2 Stimulus/Response Sequences

This feature requires a gatepass to be approved/ rejected or when the student checks-out / checks-in from the campus.

4.1.3 Functional Requirements

- REQ-1: Internet connection
- REQ-2: University's local server must be up.
- REQ-3: Firebase integration in the gatepass system.

4.4 Notification for Library

4.1.1 Description and Priority

This feature notifies user when they issued a book, and also reminds users when the return date is close.

Priority: 8

4.1.2 Stimulus/Response Sequences

Response sequence:

This feature requires a issue of book, return of book, or when day of return of book is close.

4.1.3 Functional Requirements

- REQ-1: Internet connection
- REQ-2: Library KOHA server must be up.

4.5 Query book availability in library

4.1.1 Description and Priority

This feature helps student to search for the number of copies present in library of a specific book.

Priority: 7

4.1.2 Stimulus/Response Sequences

Response sequence:

1. Select Query book
2. Type book name
3. Get results

4.1.3 Functional Requirements

REQ-1: Working UI in the mobile device

REQ-2: Internet connection

REQ-3: Library KOHA server must be up.

4.6 Show Timetable

4.1.1 Description and Priority

This is a module inspired by another similar app, Intensio. This feature takes timetable input from Nucleus API once, stores in local database and then displays the data in a good intuitive UI.

Priority: 9

4.1.2 Stimulus/Response Sequences

Response sequence:

1. Open the application
2. Done (This module is decided to be the home page of the app)

4.1.3 Functional Requirements

REQ-1: Working UI in the mobile device

REQ-2: Internet connection (for first time only)

4.7 Show attendance

4.1.1 Description and Priority

This feature is non-functional feature, which shows the attendance of the student in the recent term only.

Priority: 5

4.1.2 Stimulus/Response Sequences

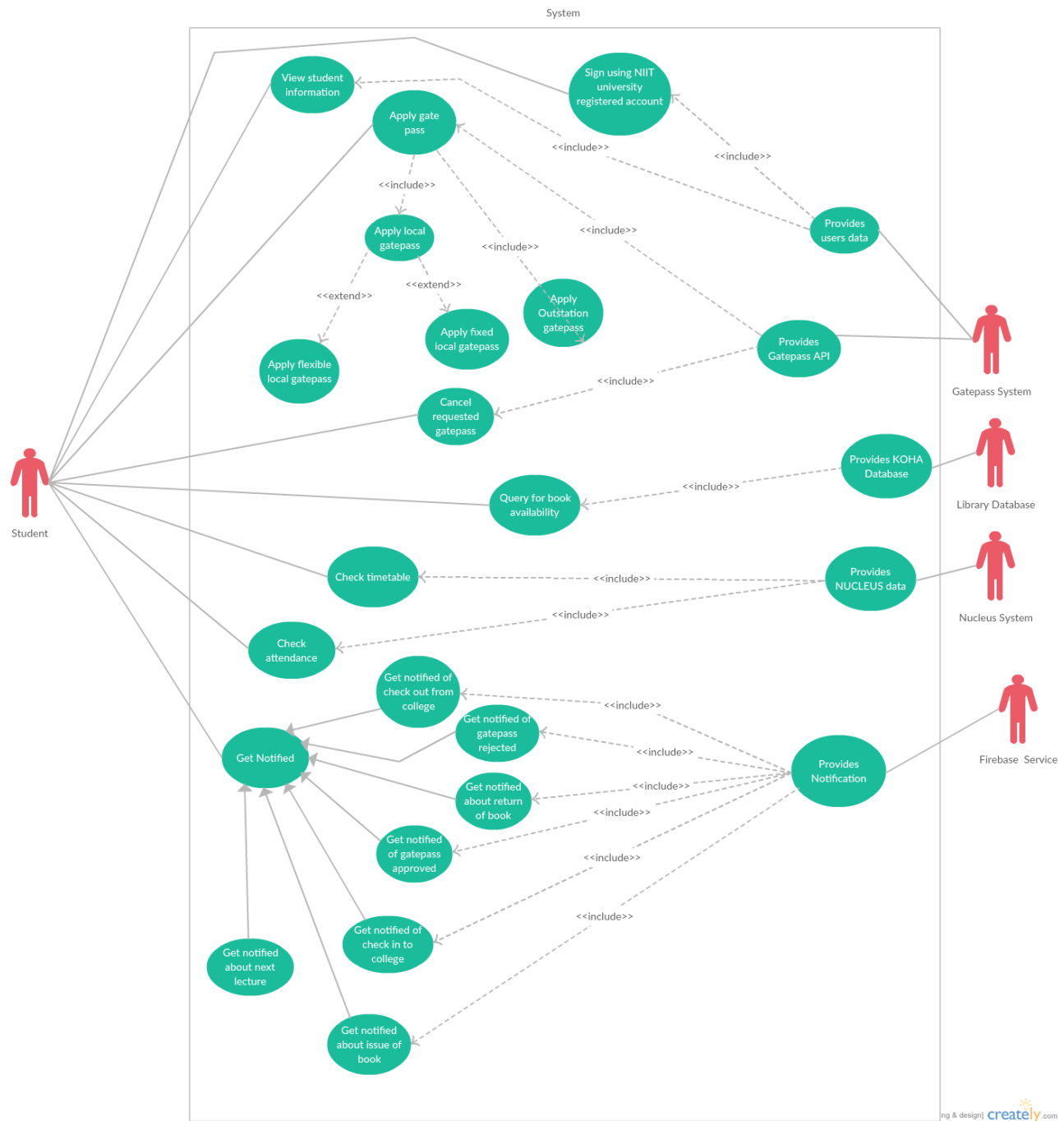
Response sequence:

1. Select View attendance
2. Done. Will see offline old attendance if internet not present.

4.1.3 Functional Requirements

- REQ-1: Working UI in the mobile device
 REQ-2: Internet connection
 REQ-3: Nucleus server must be up.

Use Case Diagram:



Use Cases List:

Use case Id	Name
UC1	View student information
UC2	Sign In to NIIT University account
UC3	Apply gate pass
UC4	Apply local gatepass
UC5	Apply flexible local gatepass
UC6	Apply fixed local gatepass
UC7	Apply outstation gatepass
UC8	Provides user's data
UC9	Cancel requested gatepass
UC10	Provides Gatepass API
UC11	Query for book availability
UC12	Provides KOHA Database
UC13	Check Timetable
UC14	Provides Nucleus data
UC15	Check attendance
UC16	Get Notified
UC17	Get notified of checkout
UC18	Get notified of checkin
UC19	Get notified of book issue
UC20	Get notified of book return
UC21	Get notified of gatepass approved
UC22	Get notified of next lecture
UC23	Get notified of gatepass rejected
UC24	Provides notification

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The app must run smoothly in 2G network. And app should also have offline support.

App must load within 1 seconds

College server data must load within 0.3 seconds as the data is present locally.

5.2 Safety Requirements

Here we need to mention the safety issues that we need to take in consideration while building, maintenance and running of our app. In our all the three modules we need to ensure with MD-5 encryption that none of the database of ERP or MOODLE servers get damaged or leaked or do some invalid options that they are not meant to do so. We do provide invalid login as safety moves if anyone with his/her entry not present in the database of ST account, tries to login in our app.

5.3 Security Requirements

First thing is our login option in our app. To do so we used Firebase as our login authentication with IONIC-2 framework. This provides securities which makes sure that only authorized users will login successfully and not the other way round. Our app makes regular API calls as database queries to our Moodle or ERP database, So we need to ensure proper securities enabled like data retrieval in JSON with encryption MD-5.

5.4 Software Quality Attributes

Here we may add some burger menu in our app which may include an option to remember Username and Passwords that can suffice. Besides this it may include some features like Bar Code Scanner to avoid the manual use of Identity cards. Other features to be think of may include some offline data synchronization to avoid use of data packs all time, hence making the app easy to use and providing Robustness. Also we can have a help menu in our burger bar to provide some helps on "How to use the App".

5.5 Business Rules

In general our users are all the students at NU. So they are the ones who will use the apps and generate DB Queries frequently. But in case of failure of some logins or DB queries , TCO or related server managers will be needed to enforce the correctness of our app at backend.

6. Other Requirements

More or less all requirements are included in the SRS. Besides all, considering about database requirements we need to have access provided by NU for making our app run, considering about the internationalization requirements we have nothing to include much since we made our app run in English and since the app is made to be run in NU so there are no regional preferences and sections involved and now considering about the legal requirements, here we need to have some legal policies to be incorporated by NU.

Appendix A: Glossary

IONIC-2 : A framework to build Web Applications.

MD-5 : Type of Encryption

HTTP : Hyper Text Transfer Protocol

FTP : File Transfer Protocol

Cordova : Apache plugins for IONIC framework

CLI : Command Line Interface

Firebase : A Google cloud integration as BaaS

Appendix B: Analysis Models

For knowing about the analysis models, refer to the attached Use-Case diagram.

Appendix C: To Be Determined List