Software Requirements Specification

For

NUFORALL

Version 1.0 approved

Prepared by:
Mohit Raj
NVS Abhilash
Prateek Agarwal
Siddharth Shashikar
Vishal Mohankumar

NIIT UNIVERSITY

September 14, 2016

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

Introduction

1.1 Purpose

The purpose of the project is to integrate the three services that are Gate-pass, Library and the Nucleus of NIIT University in one place so that the user can access them more easily and conveniently. We are building a mobile application to help the students of NIIT University so that

- The students should be able to view their current timetable with ease.
- Should be able to apply gate-pass through a mobile application itself.
- It must notify the student and concerned parent / guardian about the same.
- Should be able to view their issued books, to be notified about the reissue date.
- Student should be able to search for specific book availability in the library.

1.1 Document Conventions

- Arial font will be used in the document.
- Appropriate bullets are for describing or pointing out different features of the product.
- Font size used is 14 throughout the document.
- The headings are **bold** and the contents are normal.
- Several important phrases and words are <u>underlined</u> and **bold** for grabbing attention.
- Word spacing is same throughout and there is a line gap of 1.15 units of line spacing after each sub topic.

1.2 Intended Audience and Reading Suggestion

This document is mainly intended for anyone who is interested to know more about the project. But mainly this SRS document targets our client i.e.

NIIT University, Developers, Product Owner. For Clients and the Product Owner they should follow the index for reading this document and for the Developers they skip the initial part that is the “Introduction” of this SRS document and straight go to the “Overall Description” that is the second part of the document.

1.3 Product Scope

The product is a ***Hybrid Mobile Application*** for the students of NIIT University to improve the current system by making it more simple, reliable and handy. The App will bring together the readily used functionalities by the students of NU.

Benefits:

- The Current system will be made very handy; the users can check their daily required features in a mobile Application.
- Reliable system. Can help in last minute Gate pass.
- Check attendance in a click
- Help in avoiding penalty for late book submission.
- Will function in all platforms.

What the Application is not?

- This app is not a new system or replacement. It is intended to simplify the current process.

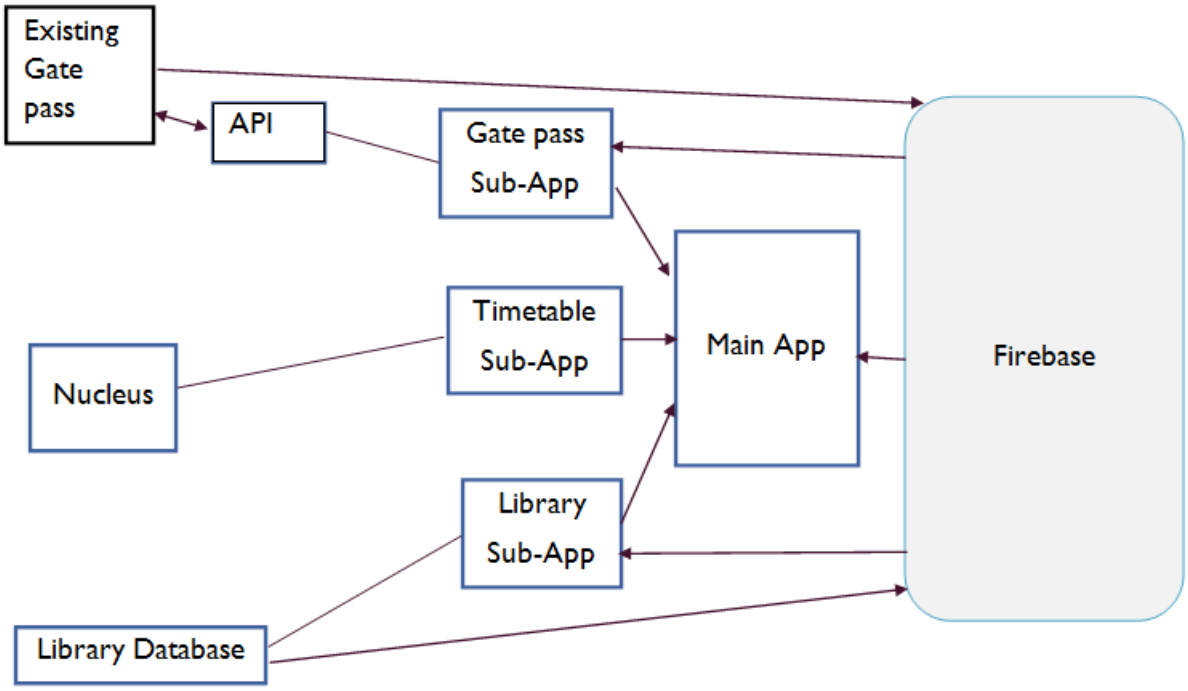
1.4 References

To be decided

2. Overall Description

2.1 Product Perspective.

This Application is a new developed mobile application that is a simplified model of the current system that is developed for the students so as to make the current system more approachable and handy. The App is a sub product of a larger system that has been simplified as per the priority needs of the students on the daily basis. The App aims only to project those features that are applicable on daily basis. The App does not replicate the UI/UX of the current system, but to produce a simpler application to access the features that are being repeatedly in use.



2.2 Product Functions

Following are the 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 gate pass 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 Application has a single user class i.e. 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 Gate pas 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

A set of visual instructions will be made available in the application itself. This application doesn't require an explicit User manual because the students are already familiar with the system and the application main motive is to simplify the process, which will surely be reflected in the simplicity of the app.

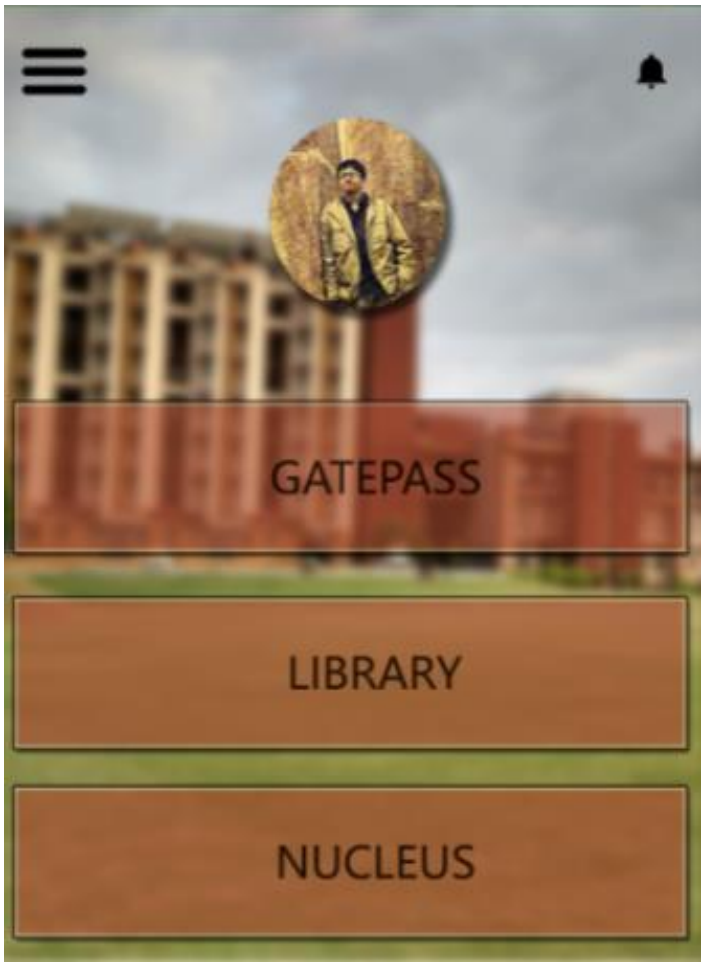
2.7 Assumptions and Dependencies

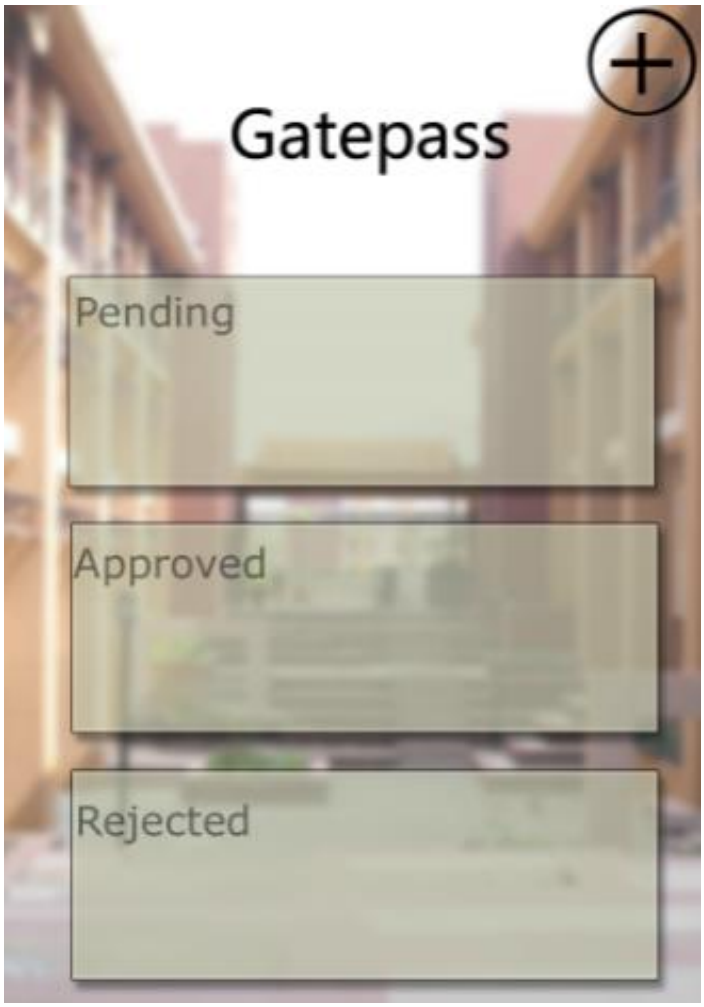
- Developers know they might not get full access to the services resulting in limiting functionality.
- Developers as of making of this document are not sure how to access the Library Koha Database, as the Kona API is not well written. They might have to directly connect to the Koha Database, which limits 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

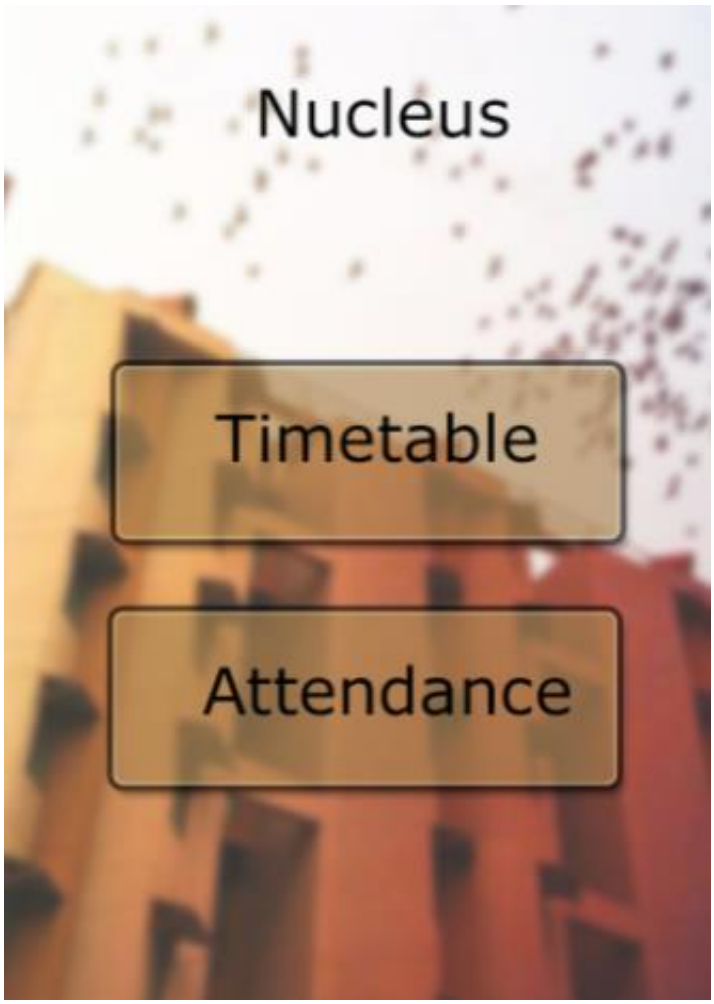
3.1 User Interfaces

Below is the prototype of the application, although this was made for getting an idea about the application and is not final


GATEPASS
LIBRARY
NUCLEUS


Gatepass
Pending
Approved
Rejected


Library
search
books
issued date
due date



3.2 **Hardware Interfaces**

The hardware to run this Application is very generic, as this is a web App this App can be deployed in any platform, i.e. Android, IOS and Microsoft Windows. Another type of hardware is access to NU database, NU servers, and cloud server. API call is used to retrieve the data from database and servers Software.

3.3 **Software Interfaces**

The application connects with the following interfaces:

1. Shall connect to the Gate pass 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 gate pass details.
5. Shall have local storage to store the timetable and other details of the app.

3.4 **Communications Interfaces**

Email notifications and push notification are enabled. Connection with servers. All these connections with firebase and servers need a reliable communication that can be achieved through network protocol HTTP and sometimes FTP Database exchange of information exchange will be made in HTTP with MD-5 encryption to make it secure.

4. System Features

Identifier	Short Name	Short Description
REQ1	Apply Local Gate pass	*This feature aims to provide students to apply local gate pass in one go. This feature helps students to remove all the overhead to log in to moodle and then open gate pass module, go to local gate pass and then apply for gate pass.*
REQ 2	Apply Outstation Gate pass	*This feature helps the user to apply a outstation gate pass with ease and removes the same overhead*

Identifier	Short Name	Short Description
		described in REQ1
REQ 3	Notifications for gate pass	*This feature is responsible to send notification to the student when requested gate pass is approved/ rejected, when student checks out and checks in.*
REQ 4	Notification for library	*This feature notifies user when they issued a book, and also reminds users when the return date is close.*
REQ 5	Query book availability in Library.	This feature helps student to search for the number of copies present in library of a specific book.
REQ 6	View Timetable	*This is a module*

Identifier	Short Name	Short Description
		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.
REQ 7	Show attendance	*This feature is non-functional feature, which shows the attendance of the student in the recent term only.*

4.1 **Apply Local Gate pass**

4.1.1 Description and Priority

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

Priority: 9

4.1.2 Stimulus/Response Sequences

Response sequence:

1. *User selects Apply Gate pass from app.*
2. *User selects Local Gate pass*
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 (gate pass blocked)
REQ-4: University's local server must be up.

4.2 **Apply Outstation Gate pass**

4.1.1 Description and Priority

.

Priority: 6

4.1.2 Stimulus/Response Sequences

Response sequence:

1. *Select Outstation gate pass*
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 (gate pass blocked)
REQ-4: University’s local server must be up.

4.3 Notifications for Gate pass

4.1.1 Description and Priority

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

Priority: 9

4.1.2 Stimulus/Response Sequences

This feature requires a gate pass 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 gate pass 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:

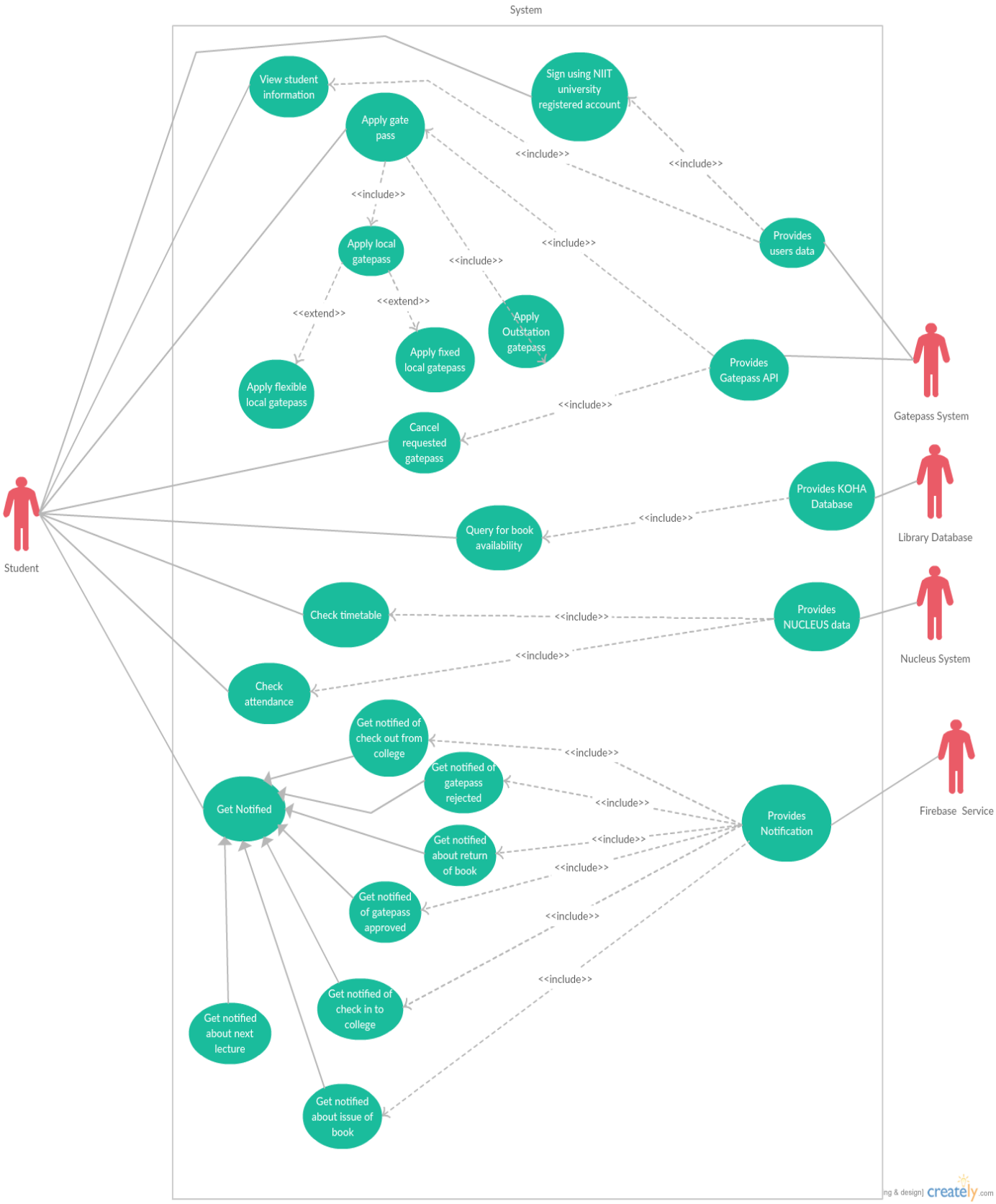
1. Select View attendance

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 Cases 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 gate pass
UC5	Apply flexible local gate pass
UC6	Apply fixed local gate pass
UC7	Apply outstation gate pass
UC8	Provides user’s data
UC9	Cancel requested gate pass
UC10	Provides Gate pass 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

Use case Id	Name
UC18	Get notified of check in
UC19	Get notified of book issue
UC20	Get notified of book return
UC21	Get notified of gate pass approved
UC22	Get notified of next lecture
UC23	Get notified of gate pass rejected
UC24	Provides notification

5. Other Nonfunctional Requirements

Performance Requirements

For the App to work fast and reliably, Ionic 2 framework along with Angular 2 have been implemented. This gives the UI/UX better look and the performance of the App is enhanced. As the tools are the current version of the market, it is more stable with lesser security issues. The App responds to the user and works with the minimum time delay.

Safety Requirements

As the App needs to constantly communicate with the NU Database and server, it is very important that safety features are taken into consideration. The App will use MD-3 Encryption to retrieve and fetch data. Bugs in the code is ensured to be kept as minimum as possible. One advantage is that this App is limited only to the students of NIIT so it will be comparatively easier to monitor the safety issues.

Security Requirements

The authentication of users is being done by Google Firebase. The at most concern is the
Data retrieval and update to the NU database and servers. MD-5 Encryption can be used to overcome the issue of someone tapping the information.

Software Quality Attributes

The App aims to deliver adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. The App will provide the users with the most robust and simple App that can help improving the current system. Many features like burger Menu, Offline synchronization, barcode scanner is used to further enhance the quality of the App

Business Rules

The developer team will be responsible for the correctness of the App. But the users will be frequently querying the NU database and server which comes under the observation of the NU TCO team. So, it will be a combined effort to minimize any distress and will be the responsibility of both the parties to ensure quality of the App.

Nonfunctional Requirements	Description
Performance Requirements	For the App to work fast and reliably, Ionic 2 framework along with Angular 2 have been implemented. This gives the UI/UX better look and the performance of the App is enhanced. As the tools are the current version of the market, it is more stable with lesser security issues. The App responds to the user and works with the minimum time delay
Safety Requirements	As the App needs to constantly communicate with the NU Database and server, it is very important that safety features are taken into consideration. The App will use MD-3 Encryption to retrieve and fetch data. Bugs in the code is ensured to be kept as minimum as possible. One advantage is that this App is limited only to the students of NIIT so it will be comparatively easier to monitor the safety issues
Security Requirements	The authentication of users is being done by Google Firebase. The at most concern is the Data retrieval and update to the NU database and servers. MD-5 Encryption can be used to overcome

	the issue of someone tapping the information.
Software Quality Attributes	The App aims to deliver adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. The App will provide the users with the most robust and simple App that can help improving the current system. Many features like burger Menu, Offline synchronization, barcode scanner is used to further enhance the quality of the App
Business Rules	The developer team will be responsible for the correctness of the App. But the users will be frequently querying the NU database and server which comes under the observation of the NU TCO team. So, it will be a combined effort to minimize any distress and will be the responsibility of both the parties to ensure quality of the App

6. Other Requirements

The entire requirement is mostly covered in the SRS. Many legal policies and requirements of the NU have to be fulfilled to get access to the NU database servers.

Appendix A: Glossary

HTTP	Hyper Text Transfer Protocol
FTP	File Transfer Protocol
Firebase	A Google cloud integration as BaaS
CRUD	Create, Retrieve, Update, Delete
CLI	Command Line Interface
Cordova	Apache plugins for IONIC framework
User	The person who operate the software product.
VScode	Visual studio code editor
CSS	Cascading style sheet
TCO	Telecommunication Organization
Koha	Backend on which library system currently works
API	Application protocol interface
Sass	Syntactically awesome style sheet
NU	Niit University

Appendix B: Analysis Models

Analysis model to be explained with the attached use-case diagram