Software Requirements Specification

Version 1.2

25th February 2016

Transport Management System

Submitted in partial fulfillment
Of the requirements of
CS 223 Software Engineering

This work is based upon the submissions of the course Software Engineering (CS223).

The students who submitted this team project were

Divya Nagar, Priyanka R. Masne, Purvi Tiwari, Rochika, Shreshtha Garg

Table of Contents

Table of Contents. i
List of Figures. ii
1.0. Introduction. 1
1.1. Purpose. 1
1.2. Scope of Project 1
1.3 Constraints. 1
1.4 Assumptions and Dependencies. 1
1.3. Glossary. 1
1.4. References. 1
1.5. Overview of Document 1
2.0. Overall Description. 2
2.1 System Environment 2
2.2 Functional Requirements Specification. 2
2.2.1 Use case 1. 2
Use case: 2
2.3 User Characteristics. 2
2.4 Non-Functional Requirements. 2
3.0. Requirements Specification. 3
3.1 Functional Requirements. 3
3.1.1 << Name of the first feature>>. 3
3.1.2 << Name of the second feature>>. 3
3.3 Detailed Non-Functional Requirements. 3
3.4 Logical Structure of the Data. 3
4.0 Supporting information. 3
4.1 Table of contents and index. 3
4.2 Appendixes. 3

List of Figures

No table of figures entries found.

1.0. Introduction

1.1. Purpose

The purpose of this document is to give a detailed description of the requirements for the “Transport Management System” software. It will illustrate the problems that the software deals with, the target users, the constraints under which it works and its features.

1.2. Scope of Project

This software will be designed to provide detailed bus schedule which will be prepared to fulfill the academic and student requirements while optimizing the number of buses used. This will make some of the processes automated which will save the energy and cost incurred in doing it manually. So the software does more work in lower cost. Furthermore, Internet is required for the usage of this software.

The input will be academic schedule given by an existing system mentioning number of classes and number of registered students for each class. The software automatically fetches the number of students catching bus while arrival and departure. Other inputs are the number of buses, capacity of each bus and cost of each trip. The output will be the number of buses at different time instances, a copy of which can be downloaded.

There will be interaction between admin and requester through web-portal. Thus there will be 2 kinds of user accounts.
All the information is maintained in database, which is located on web-server.

The software facilitates request for extra buses in cases of extra class or students’ activity requirement. The requester gives ‘required time’ and ‘number of students’ as inputs to the portal. Software calculates the number of buses required, displays the request and availability of buses to admin. In case of availability, request is granted otherwise discarded.

Also there will be a feature of notifying request approval/denial to the person who requested the bus.
An invoice for the regular and extra bus will be produced at the end of each month.

1.3 Constraints

1. Number of buses are fixed until given again as a new input and is sufficient to meet the requirements.
2. Capacity of a bus is fixed until given again as a new input.
3. The academic schedule and bus schedule are followed strictly.

1.4 Assumptions and Dependencies

1. All the courses mentioned in academic schedule are compulsory.
2. Every student attends all compulsory classes.
3. No instantaneous allocation of bus is considered.
4. The software is dependent on the system providing academic schedule as input.

1.3. Glossary

Term	Definition
User	Someone who is interacting with the software
web-portal	A web application through which requests are handled
Requestor	Someone who is making a request. Here they can be either faculty or Student Representatives.
Admin	Someone who is responsible for approval and denial of requests. Here admin is our user also.

1.4. References

IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.* IEEE Computer Society, 1998.

1.5. Overview of Document

The rest of the document is designed in the following way:

The first chapter described the purpose of document, scope of the project describing features in brief, constraints and assumptions .

The second chapter describes the user characteristics i.e. the types and nature of the users. Also non functional requirements are highlighted such as security, reliability, performance requirements etc.

The third chapter illustrates the requirements such as system environment, functional requirements that the software should meet. It describes the features in more elaborated manner with each of its use cases.

2.0. Overall Description

2.1 System Environment

<< Keep blank for the time being >>

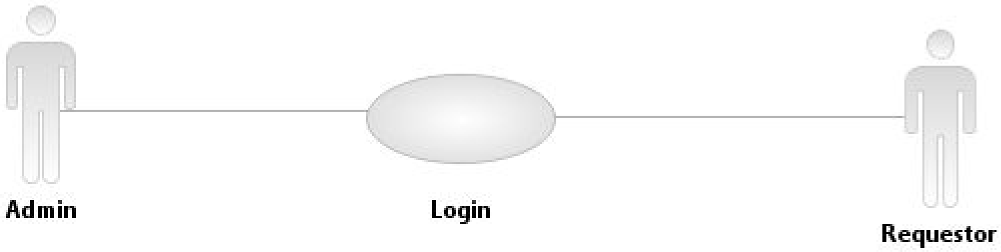
2.2 Functional Requirements Specification

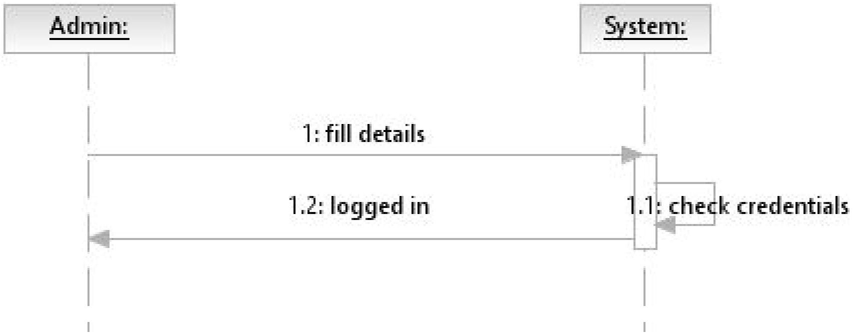
<< Keep blank for the time being >>

2.2.1 Use case diagrams

Use case : User Login

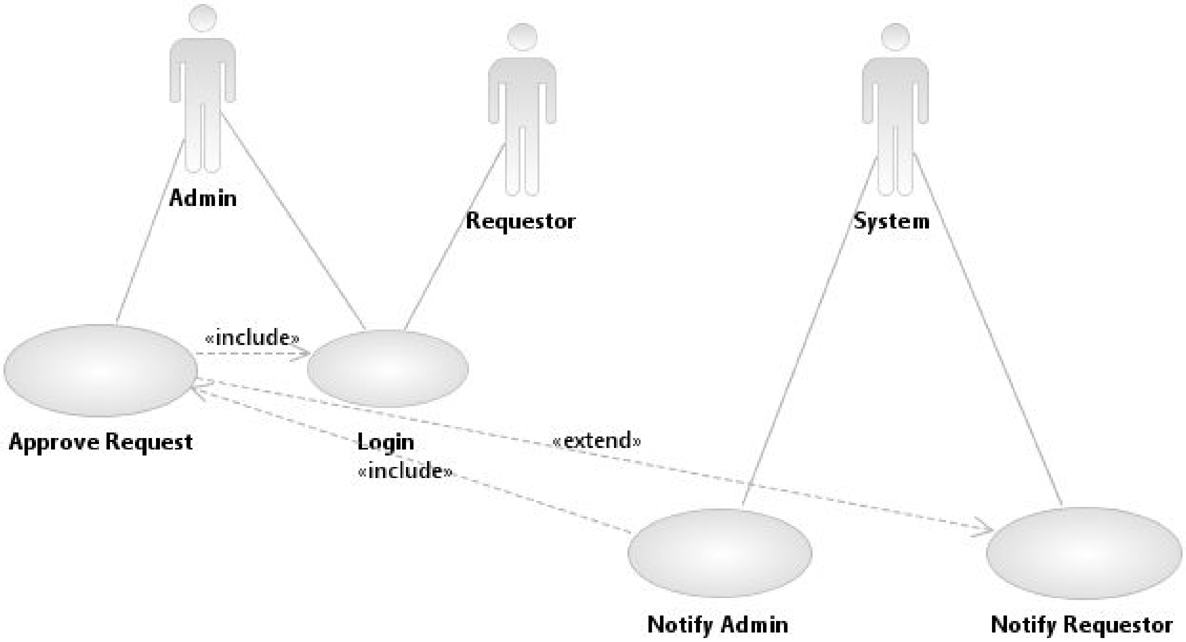
Diagram:

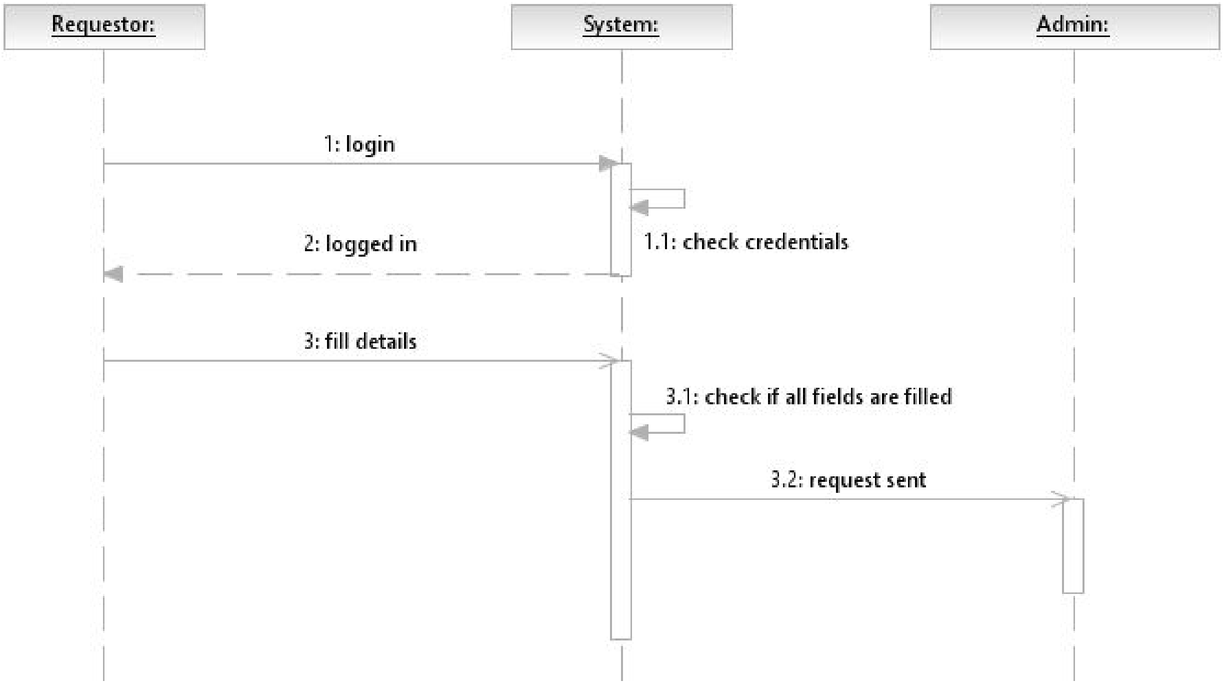




Use case: Approve request

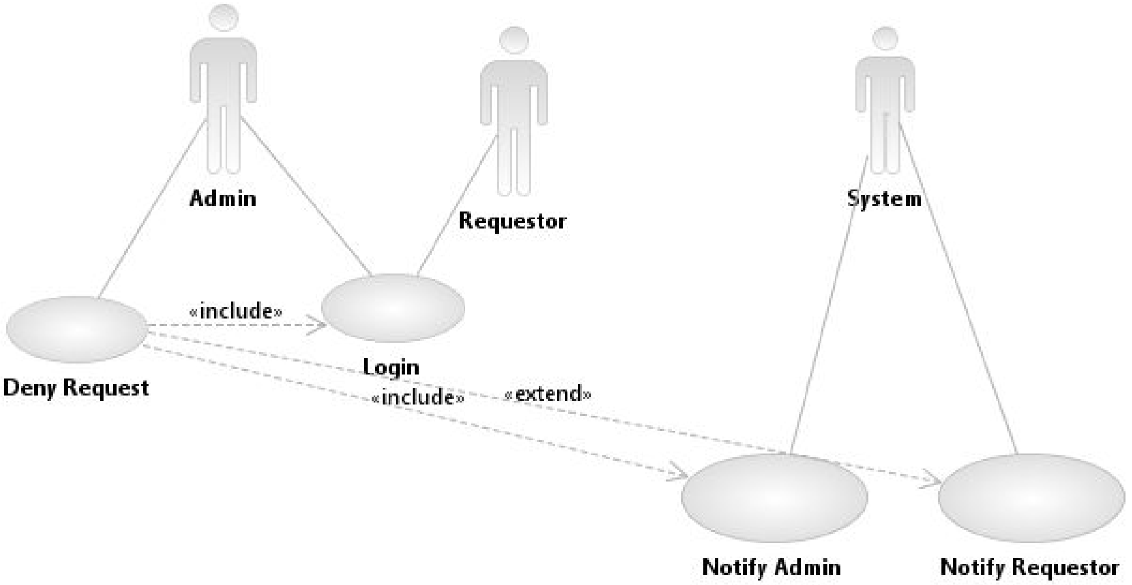
Diagram:

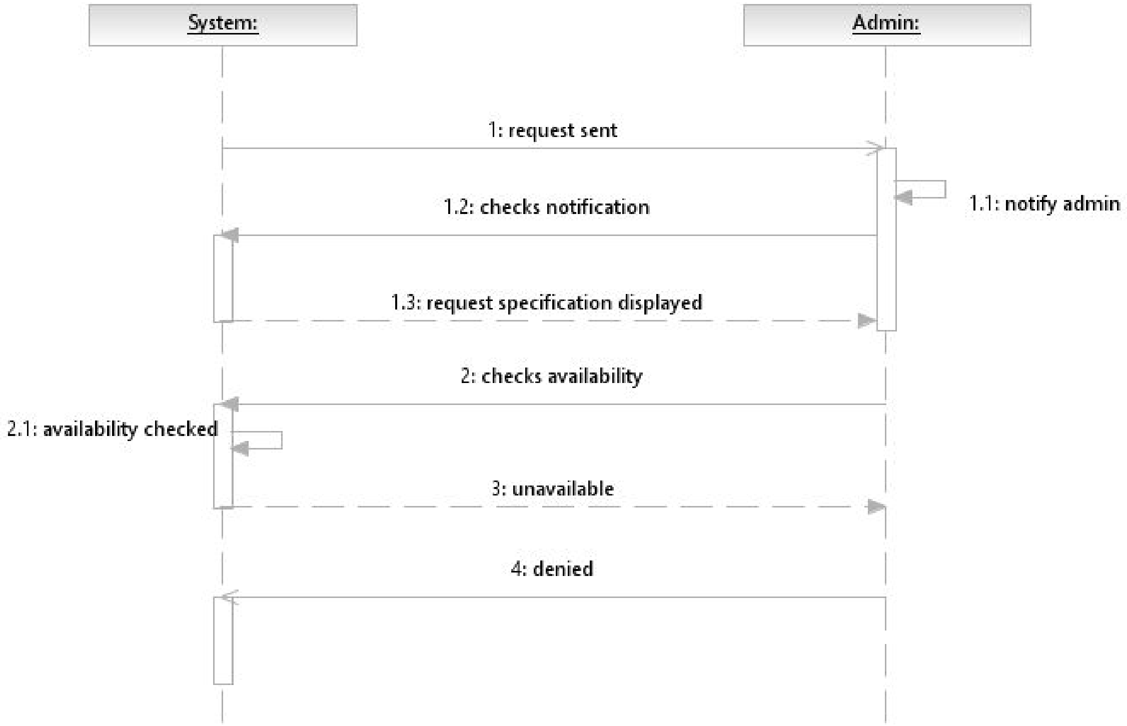




Use case: Deny Request

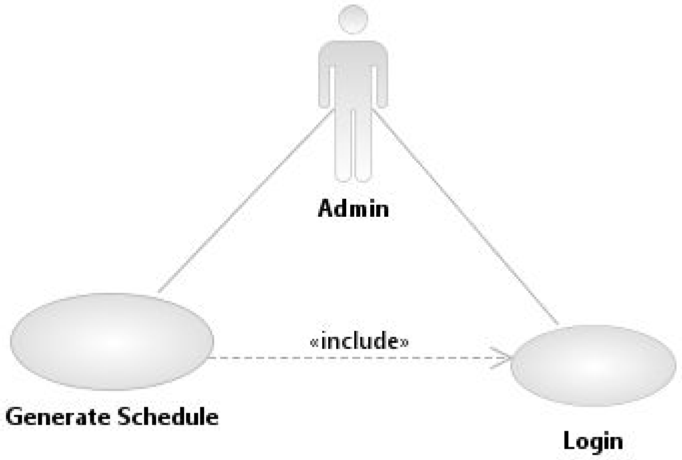
Diagram:

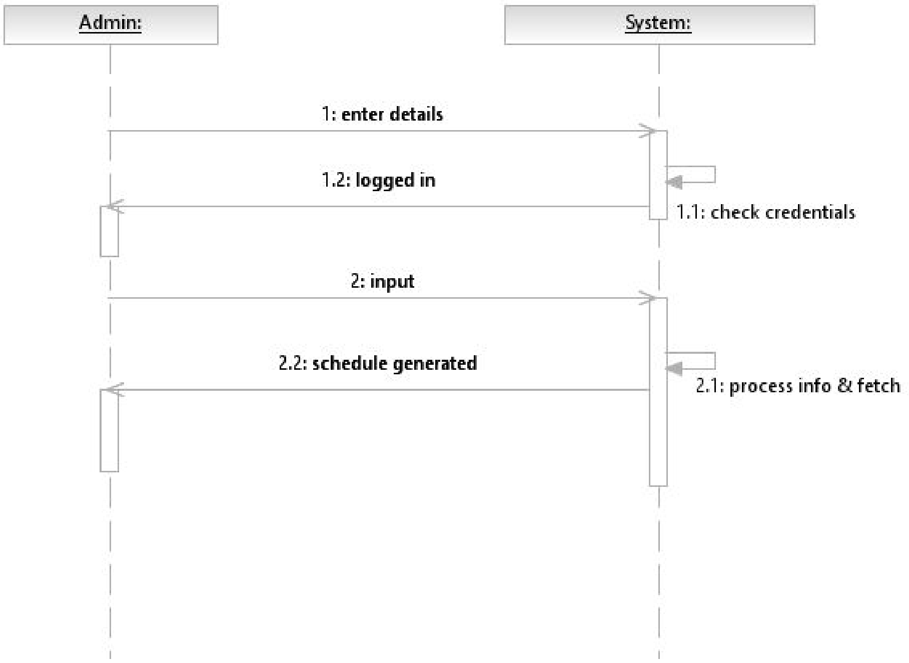




Use case: Generate Schedule

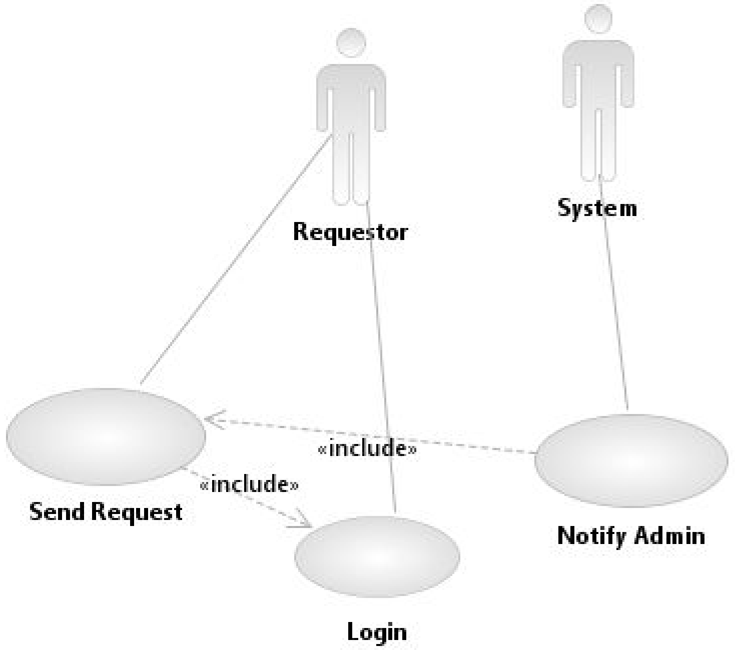
Diagram:

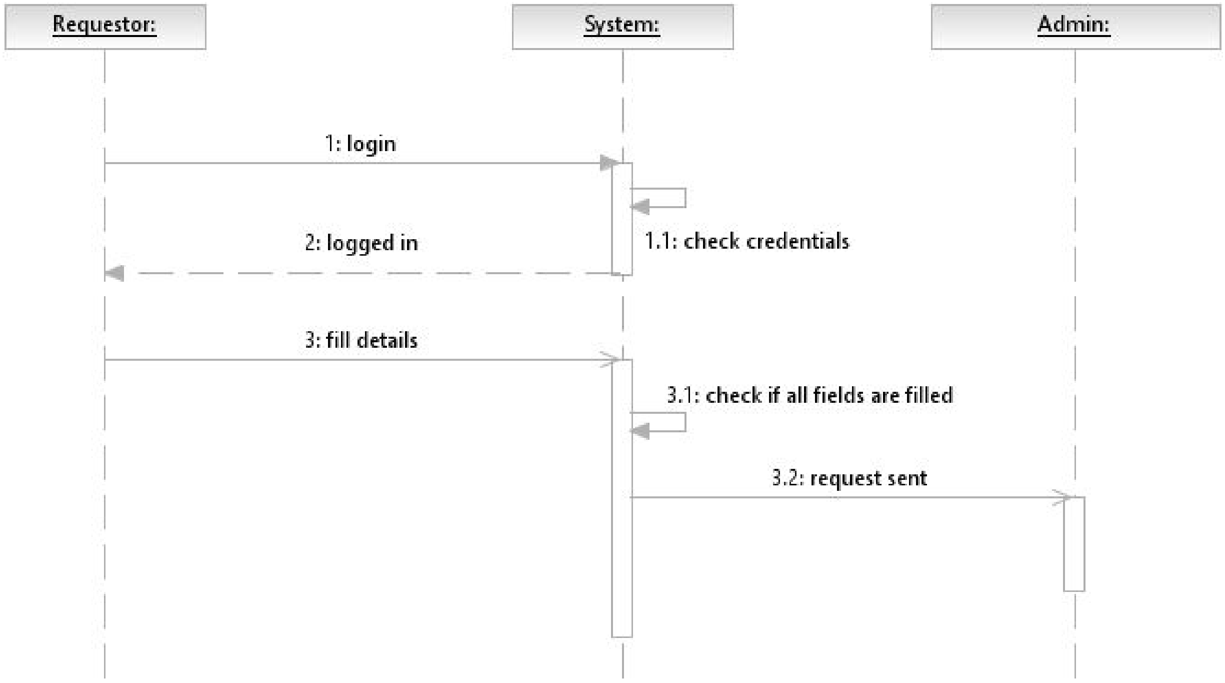




Use case: Send Request

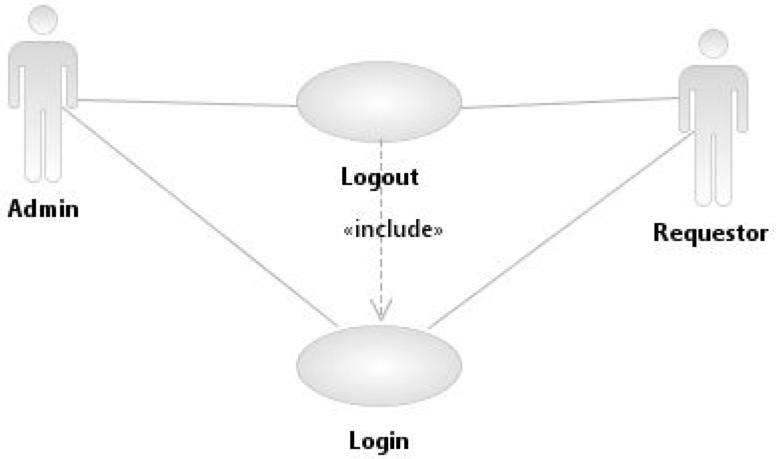
Diagram:

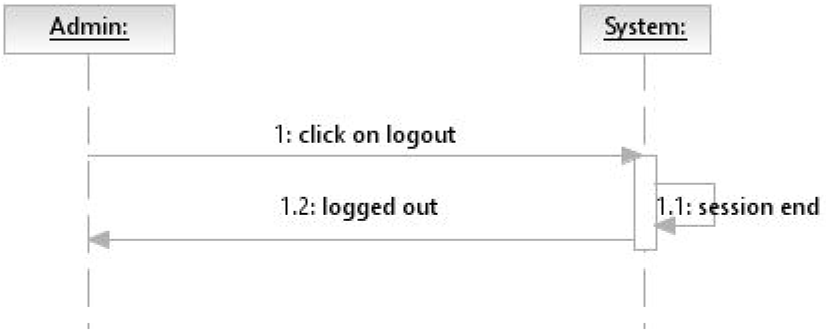




Use case: Logout

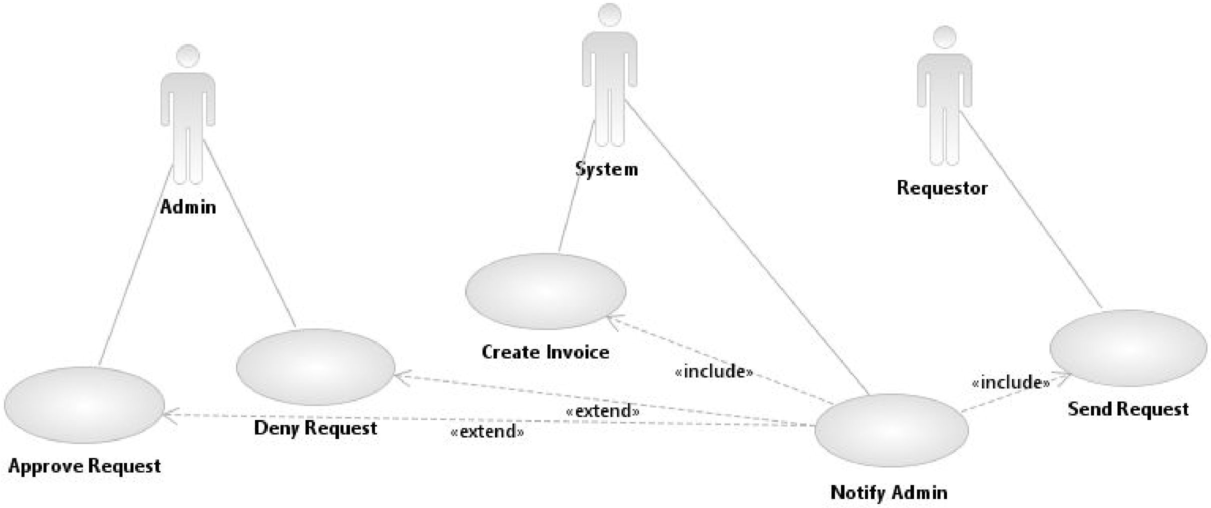
Diagram:

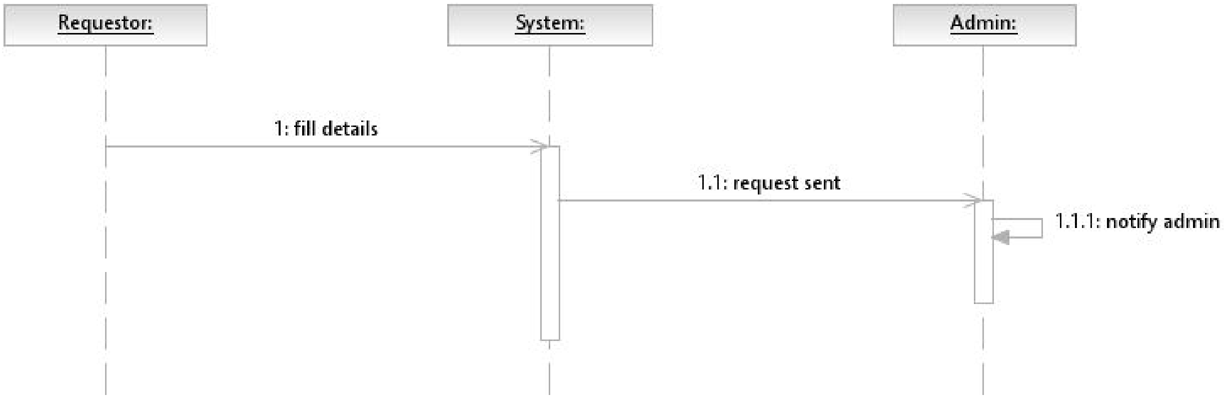




Use case: Notify Admin

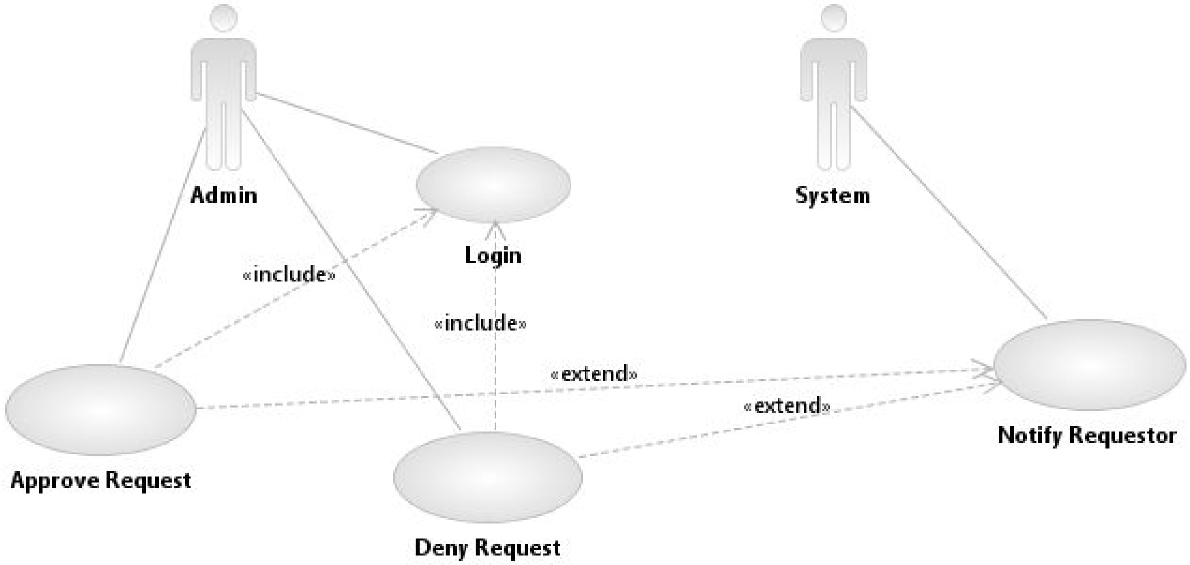
Diagram:

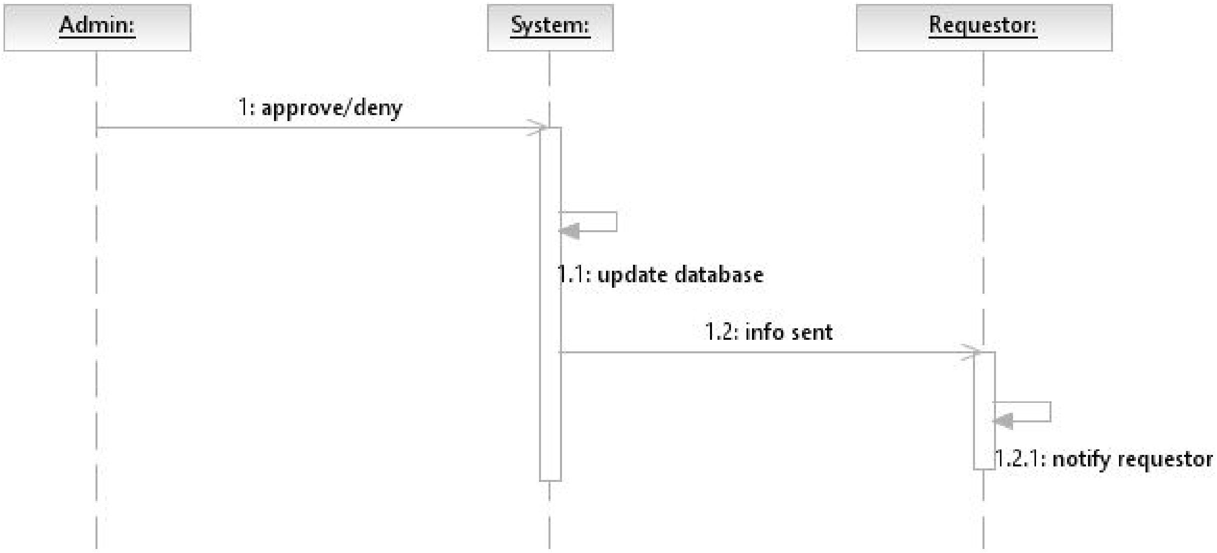




Use case: Notify Requestor

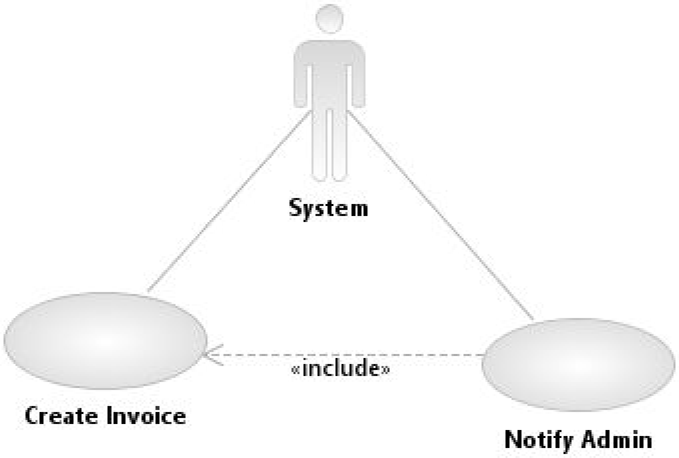
Diagram:

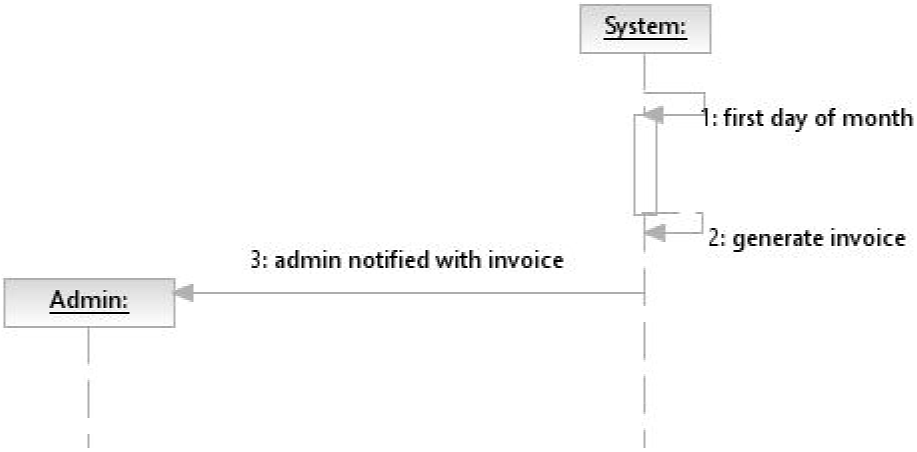




Use case: Create Invoice

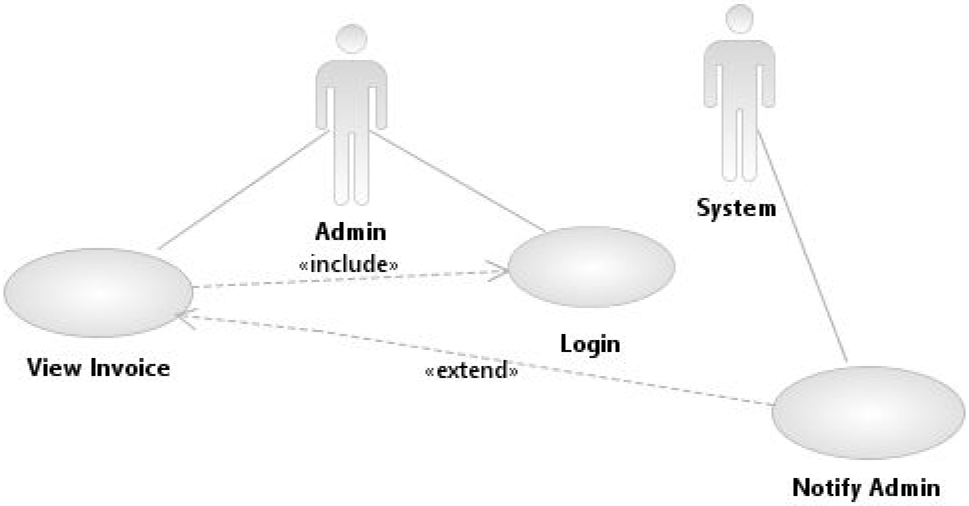
Diagram:

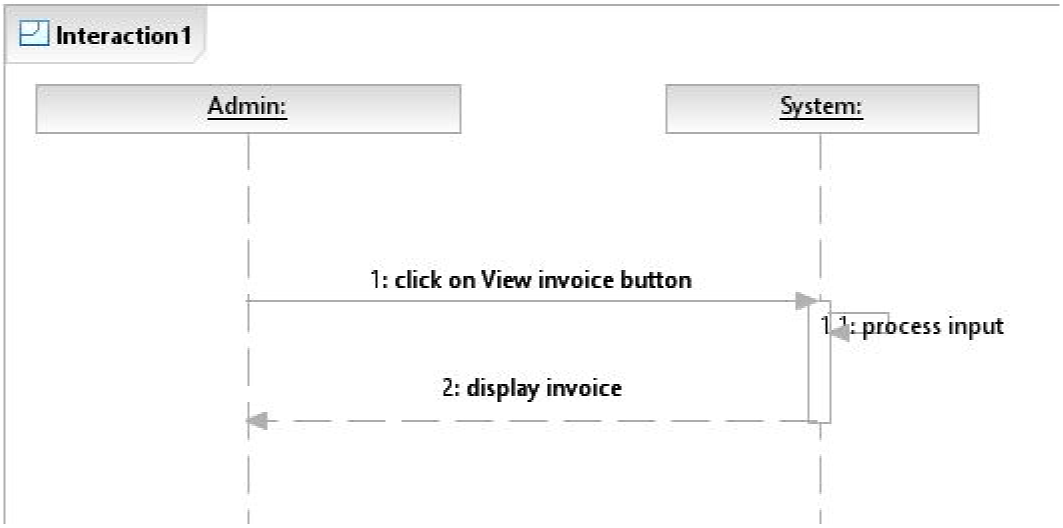




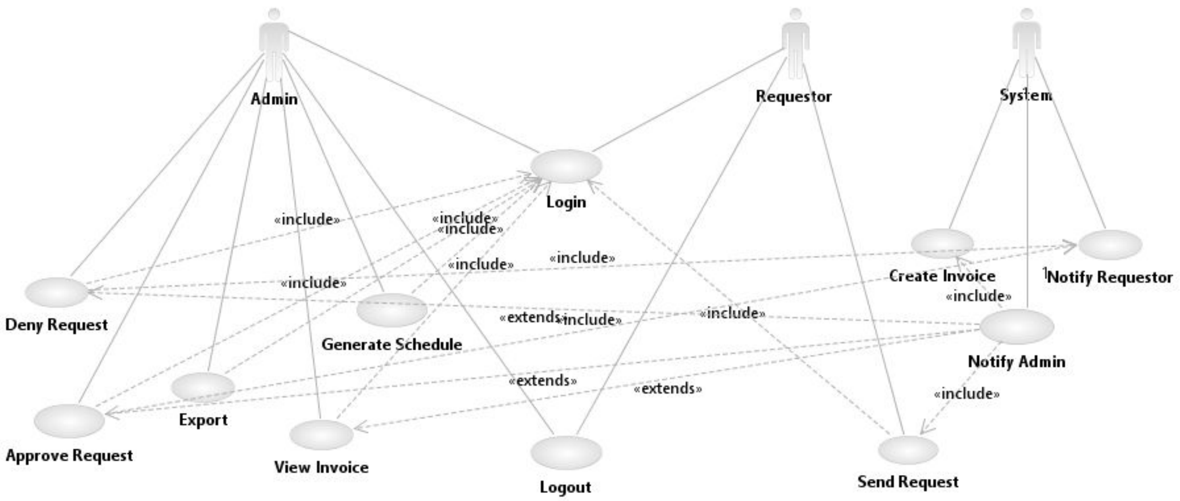
Use case: View Invoice

Diagram:

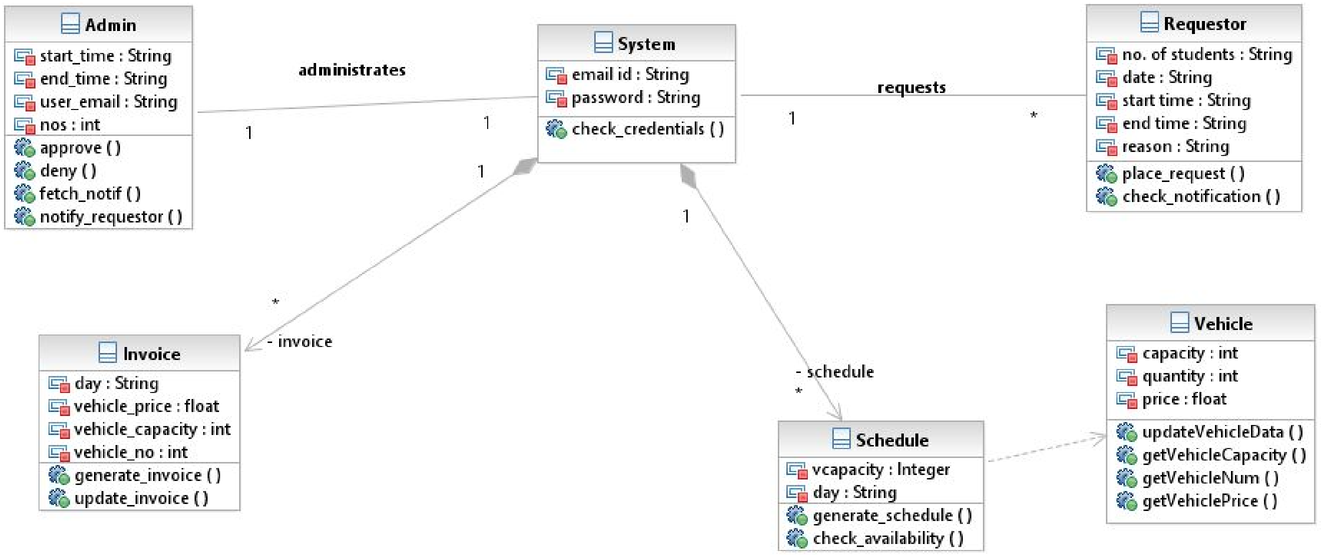




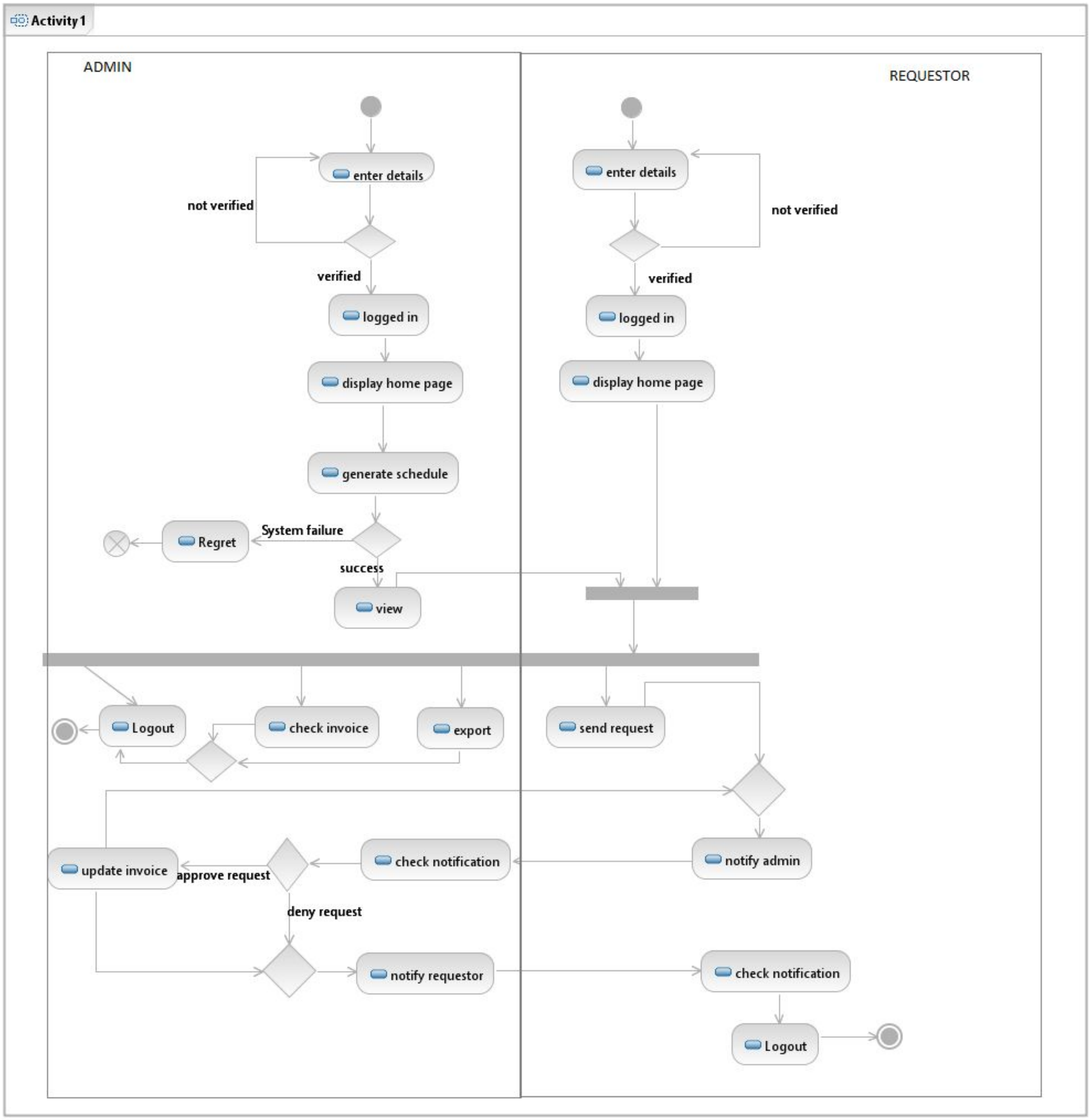
USE CASE DIAGRAM



CLASS DIAGRAM



ACTIVITY DIAGRAM



Brief Description

Initial Step-By-Step Description

2.3 User Characteristics

The users should have a hand on computers and is aware of internet applications.
There are two kinds of users interacting with this software.
1.Users who will use the portal and request for extra buses.
2.Users who after taking the inputs, will run the software and generate the bus schedule to be displayed. This user also administers the requests for extra buses on portal.

2.4 Non-Functional Requirements

1. User Interface Requirements -The user interface of the application must be user-friendly, intuitive and easy to use.

2. Performance requirements - The system shall function in real-time: any operation on the stored information, shall complete in less than 10 seconds.

3. Reliability – The software works only in the presence of Internet. Although the final generated schedule can be exported and saved.

4. Security - The software takes users user-id and password and hence only the approved credentials can take benefit of request feature.

3.0. Requirements Specification

3.1 Functional Requirements

3.1.1 User Login

Use Case Name	User login
Trigger	Clicking on “Login” or “Submit” button
Precondition	User should have signed up
Basic Path	1.Visiting the website 2. Log in with credentials
Alternative Paths	NA
Postcondition	User with correct credential is allowed to login, otherwise error
Exception Paths	1.If connection is terminated or the credentials are wrongly filled, then all fields will again become blank 2.User has not signed up
Other	NA

3.1.2 Send Request

Use Case Name	Send request
Trigger	“Request” button is clicked
Precondition	User should be logged in

Basic Path	1.Click on request button 2.Fill inputs required for the request 3.Click the request button
Alternative Paths	NA
Postcondition	Request is made if all compulsory fields are filled otherwise error.
Exception Paths	Admin has no such feature.
Other	NA

3.1.3 Notify Admin

Use Case Name	Notify Admin
Trigger	Whenever request is made by requester
Precondition	Request is pending on server
Basic Path	The admin logs in the portal
Alternative Paths	NA
Postcondition	Admin receives a notification
Exception Paths	Requester does not have this feature
Other	NA

3.1.4 Approve Request

Use Case Name	Approve Request
Trigger	“Approve” button is clicked
Precondition	Buses are available at that time
Basic Path	1.Admin logs in the portal and checks notification 2.Admin checks the availability of buses 3. Clicks the approve button

Alternative Paths	NA
Postcondition	1. Requester is notified 2. Extra bus added in invoice of respective date 3. Availability is updated
Exception Paths	NA
Other	NA

3.1.5 Deny Request

Use Case Name	Deny Request
Trigger	“Deny” button is clicked
Precondition	No bus availability at that time
Basic Path	1.Admin logs in the portal and checks notification 2.Admin checks the availability of buses 3.Clicks the deny button
Alternative Paths	NA
Postcondition	Requester is notified
Exception Paths	NA
Other	NA

3.1.6 Notify Requestor

Use Case Name	Notify Requestor
Trigger	Whenever request is approved/denied by admin
Precondition	Request is pending for approval
Basic Path	1.The admin logs in the portal 2.Admin checks availability 3.Admin approves or denies

Alternative Paths	NA
Postcondition	Requester receives a notification
Exception Paths	Admin does not have this feature
Other	NA

3.1.7 Schedule Generation

Use Case Name	Schedule Generation
Trigger	“Generate” button is clicked
Precondition	The academic schedule has been taken as input in required form from the other system.
Basic Path	1 .Open the software 2. Fill in the academic schedule, number of buses, capacity of each bus and cost of one round trip as input 3. Click the generate button 4.The software will automatically extract the number of students attending each class 5. It will accordingly divide the number of students catching bus at all time intervals 6. Number of buses required as per bus capacity are allocated at their respective time instances
Alternative Paths	NA
Postcondition	Bus schedule is successfully generated and displayed
Exception Paths	Power failure might interrupt the generation process and displays error
Other	NA

3.1.8 Create Invoice

Use Case Name	Create Invoice
Trigger	Last day of month
Precondition	The date and count is reset on 1st date of every month
Basic Path	1. The clock will continuously run on server 2. On approval of each extra bus, route count will increase 3.For each extra route cost will be added 4. Total variable cost will be added to the fixed cost
Alternative Paths	NA
Postcondition	The total cost is calculated and notified to admin
Exception Paths	NA
Other	NA

3.1.9 View Invoice

Use Case Name	View Invoice
Trigger	Click on “View” button
Precondition	Admin has successfully logged in
Basic Path	Admin has to login and click on view button
Alternative Paths	NA
Postcondition	
Exception Paths	NA
Other	NA

3.1.10 Logout User

Use Case Name	Logout User
Trigger	Click on “logout” button
Precondition	User has logged in
Basic Path	User has to click on logout button
Alternative Paths	NA
Postcondition	User is logged out from the system
Exception Paths	NA
Other	NA

3.3 Detailed Non-Functional Requirements

3.4 Logical Structure of the Data

<< Keep this blank for the time being>>

4.0 Supporting information

4.1 Table of contents and index

4.2 Appendixes

Version 1.0(17th January) : added use case “Logout” and “View invoice” and removed “Sign up”.

Version 1.1(28th January) : updated Use Case Diagrams, Activity Diagram, Class Diagram and Sequence Diagram.
Version 1.2 (25th February): Updated Use case Diagrams, Class diagrams and Sequence Diagrams, Added Data Flow Diagram, Implemented Coding