RESOURCE MANAGEMENT SYSTEM

Submitted in partial fulfillment of the requirements of

CS F213 Object Oriented Programming

Priyanka M.P.	2012B5A7483G
Aditya Sunil Joshi	2012A7PS004G
Rohit Gupta	2012A8PS379G
Jatin Khandelwal	2013A7PS105G
Siddharth Gupta	2013A7PS017G
Siddharth Mohan	2013A7PS276G
Udit Guru	2011B3A8431G

Table of Contents

1.0 Introduction

1.1 Purpose

The purpose of this document is to present a detailed description of the Resource Management System. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system.

1.2 Scope of the Project

This project aims to efficiently manage resources used by students on campus via an online portal. This includes services such as cab sharing and room booking.

When it comes to the process of room booking, the system is designed to minimize manual interference and hence automate the process. The students then have the option of booking classrooms right from the convenience of their hostel rooms. There are options for booking a classroom, LT, Auditorium or VGH. After checking for availability, the system directs the requests to the concerned authorities to get the necessary permissions and finally allots the room.

In case of cab sharing, the system gives the option of booking or sharing a cab. Passengers who have agreed to share a cab are then grouped according to common source/destination/time and allotted cabs accordingly. The system also hopes to include route based grouping of passengers. This is done to reduce the total overall cost of the passenger and to also book a net profit for the service provider.

The system interacts with the user, provides necessary information and stores important data onto an already available database of students, authorities, rooms and taxis.

1.3 Glossary

Term	Definition
Student	A student of BITS Goa
Cab Admin	Responsible for adding/removing a cab; changing fare betwee routes.
Room Admin	Checking requests
SWD	Responsible for transactions from a student's account in case of successful journey.
User Database	A database of students and admins
Room Database	A database of classrooms, LTs, Auditorium and VGH showing their availability at any period of time
Cab Database	A database of cabs along with capacity, availability, etc

2.0 Overall Descriptions

2.1 System Environment

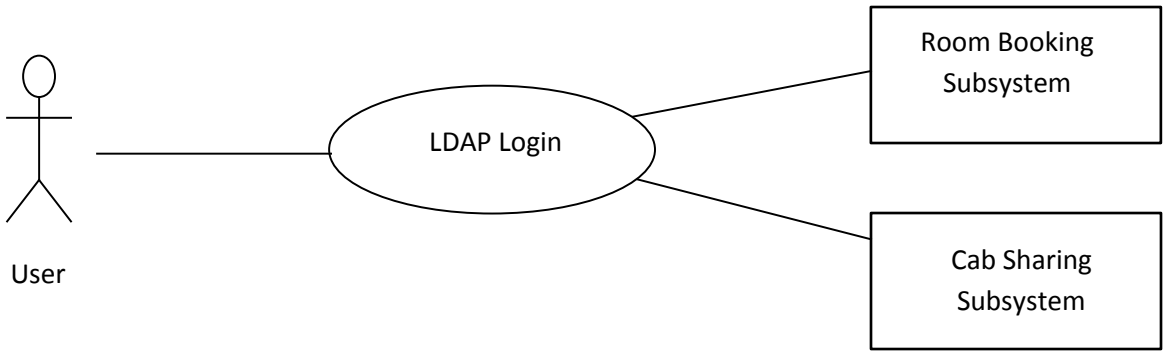
The student logs in and chooses either to book a room or share a cab. Hence the outer main system consists of two subsystems-one each for room booking and cab sharing.

In the room booking subsystem, there are five actors-student, ARC, AVU, PRO, admin and the database. A student chooses to book either a classroom, LT, Audi or a VGH room. The student then fills in necessary details along with a reason, which is validated by the concerned authorities, and finally approved for allocation. This data is then fed into the database to update the availability of the specified room.

In the cab sharing subsystem, there are four actors-passenger, SWD, admin and the database. A passenger (student) fills in details such as source, destination, time, etc and then chooses to either book or share a cab. This data is fed into the database, which groups passengers accordingly. After reserving a cab, the amount is taken from SWD. But on cancellation, part of it is refunded.

2.2 Functional Requirements Specification

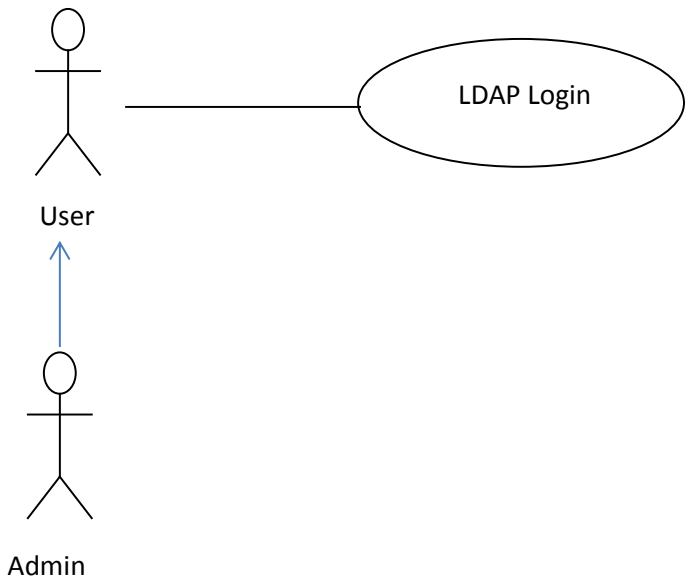
The following outlines the working of the main as well as the two subsystems.



2.2.1 Main System

User Use Case

Use case: LDAP Login
Diagram:



Brief Description: The student logs into the system using the LDAP username and password.

Initial Step-by-step Description:

1. The student first logs into the online portal.
2. Enters LDAP username (e.g. f2012483) and password.
3. An admin also logs in with a username and password which is recognized by the database as that of the admin.

3.2.2 Room Subsystem

Student Use Cases

Book a room use cases:

Use Case: Book a room

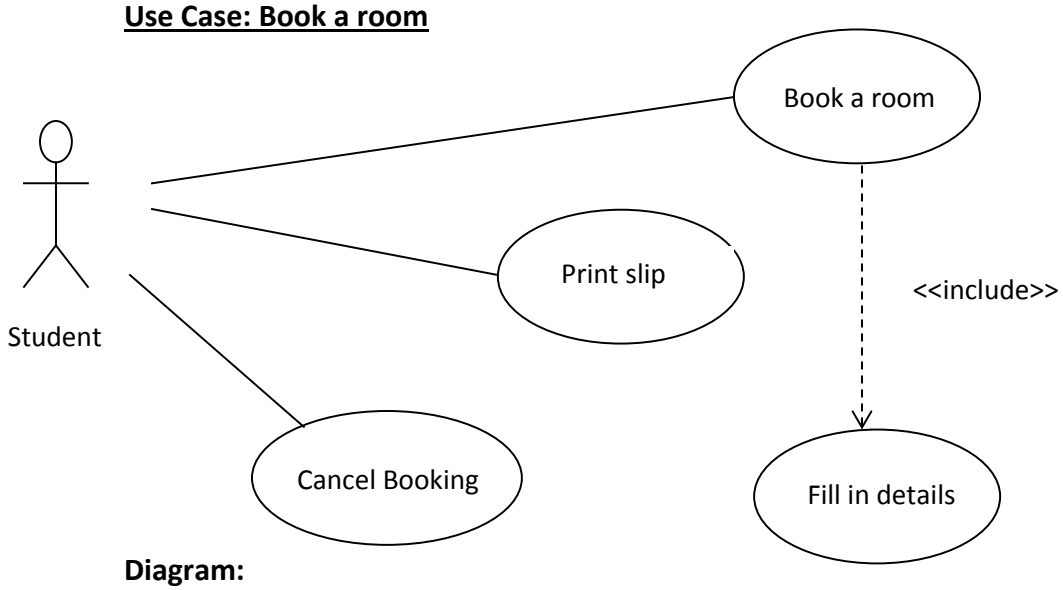
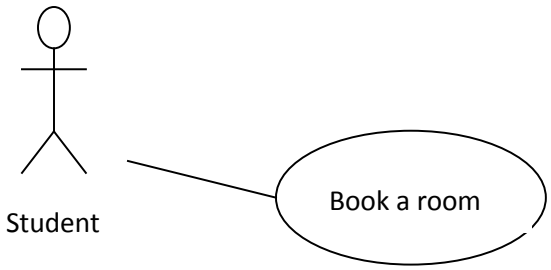


Diagram:



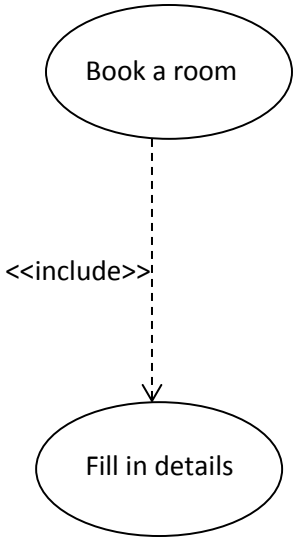
Brief Description: Student chooses a room of choice

Initial Step-by-step Description:

1. After the student successfully logs in and chooses room booking, there's a list of options to choose from.
2. After selecting the desired type, the system takes the student to the next screen.

Use Case: Fill details

Diagram:



Brief Description: Student fills in details

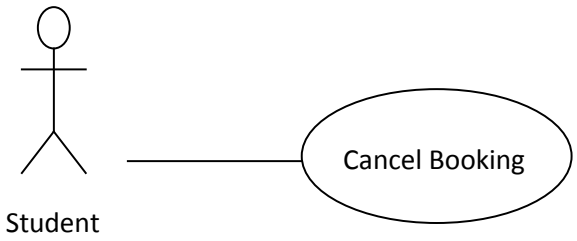
Initial Step-by-step Description:

1. After choosing the room type, the student is prompted to provide subsequent information like date, time and duration.
2. Following this, a list of available rooms at that time slot for that particular room type is displayed to the student from the database.
3. The student then chooses a room.
4. Student also has an option to ask for projectors/other technical requirements at this stage.

Cancel Use Case

Use Case: Cancel Room Booking

Diagram:



Brief Description: Student cancels the already booked room

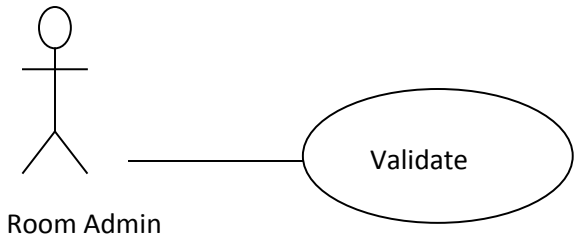
Initial Step-by-step Description:

1. Once the room has been allotted, the student has an option to cancel room booking.
2. The student should provide a valid reason for cancellation.

Room Admin Use Cases

Use Case: Validate

Diagram:

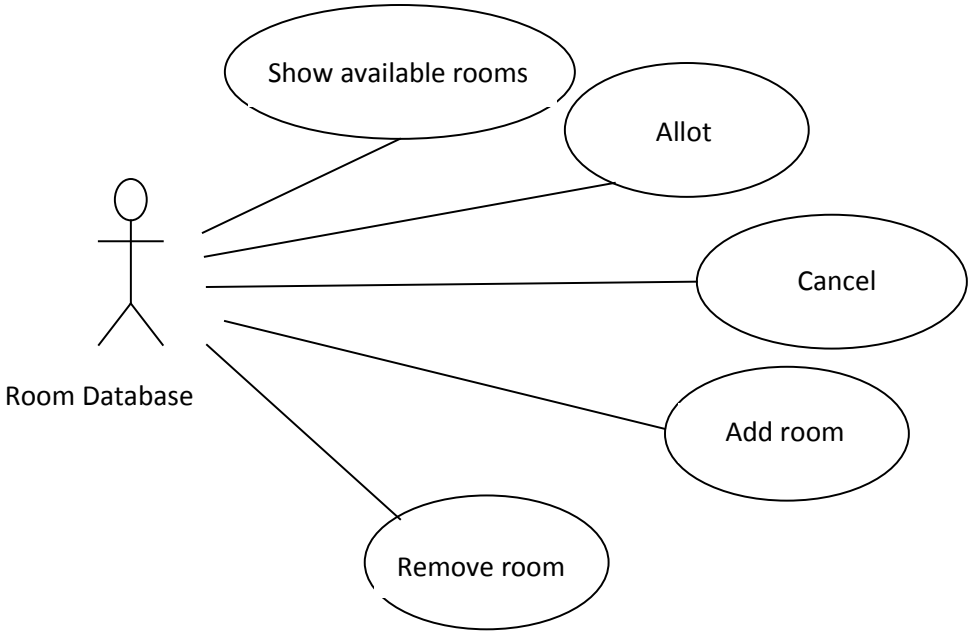


Brief Description: ARC checks the reason and can accept/reject allotment of classroom/LT

Initial Step-by-step Description:

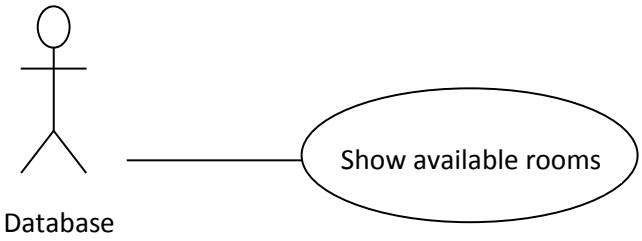
1. The room admin checks request for room allotment and approves/rejects it.

Room Database Use Cases



Use Case: Show available rooms

Diagram:



Brief Description: Database shows a list of available rooms in the particular room type

Initial Step-by-step Description:

1. After scanning through the entire list of rooms, the db shows a list of available rooms at the specified time slot and room type, to the student.

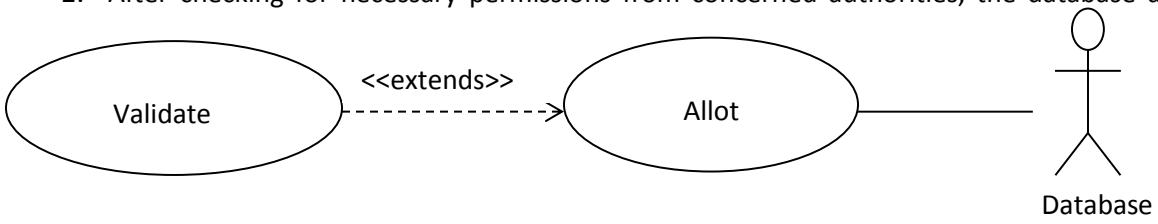
Use Case: Allot

Diagram:

Brief Description: Allots the room

Initial Step-by-step Description:

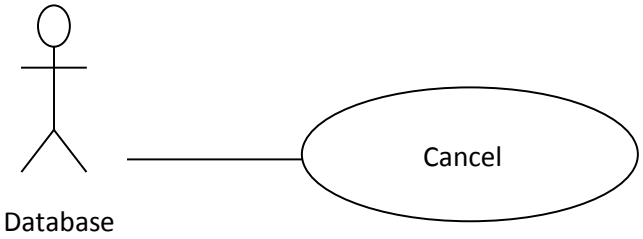
1. After checking for necessary permissions from concerned authorities, the database allots



the specified room to the student.

Use Case: Cancel

Diagram:



Brief Description: Database cancels allotted room

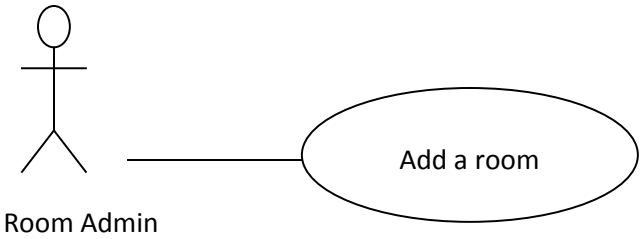
Initial Step-by-step Description:

1. If a request for cancelling the allotted room is made, the db cancels the room and sends a notification to the concerned authorities. This is to make sure that blatant cancellation of rooms is avoided.

Admin Use Cases

Use Case: Add a room

Diagram:



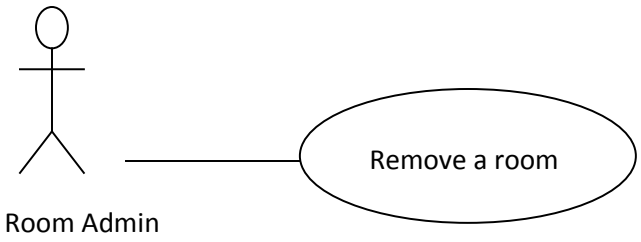
Brief Description: An extra room is added to the database

Initial Step-by-step Description:

1. Admin has special privileges to add a room to the db.

Use Case: Remove a room

Diagram:



Brief Description: A room is deleted from the database

Initial Step-by-step Description:

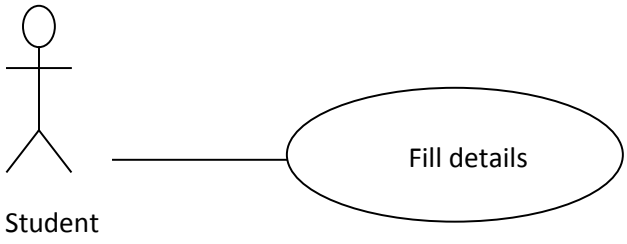
1. Admin has special privileges to remove a room to the db.

3.2.3 Cab Subsystem

Passenger use cases

Use case: Fill in details

Diagram:



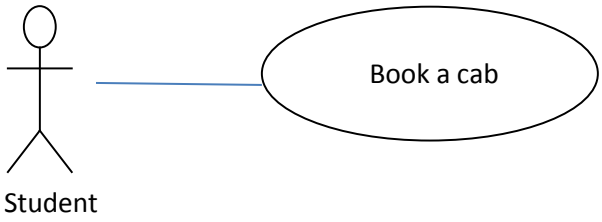
Brief Description: Passenger fills in details of the journey

Initial Step-by-step Description:

1. Passenger fills in source, destination, date, time and the number of passengers.

Use case: Reserve a cab

Diagram:



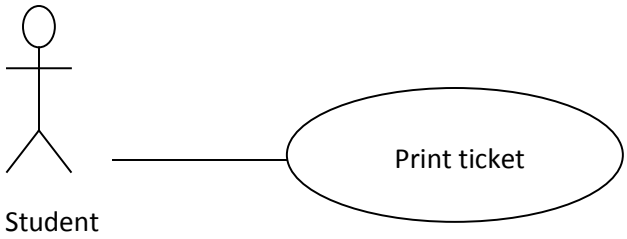
Brief Description: Passenger chooses a cab from available cabs

Initial Step-by-step Description:

1. The passenger has the option to choose cab of choice.
2. The passenger then confirms booking.

Use case: Print ticket

Diagram:



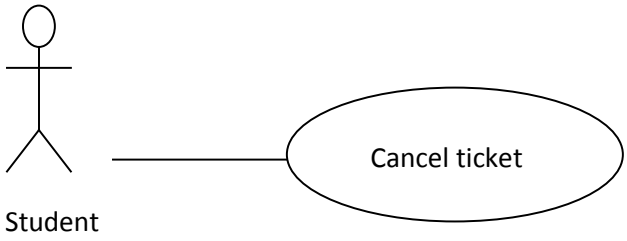
Brief Description: Passenger prints the cab ticket details if necessary

Initial Step-by-step Description:

1. Passenger prints cab ticket with taxi no., co-passengers, trip date, time, etc.

Use case: Cancel ticket

Diagram:



Brief Description: Passenger prints the cab ticket details if necessary

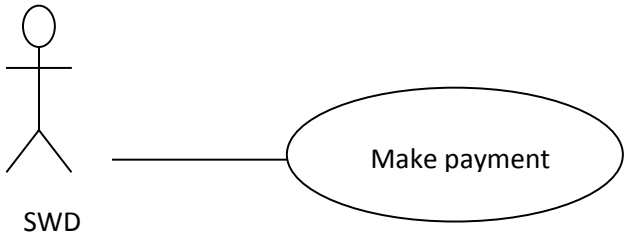
Initial Step-by-step Description:

1. Passenger prints cab ticket with taxi no., driver details, co-passengers, trip date, time, etc.

SWD Use Cases

Use case: Make Payment

Diagram:



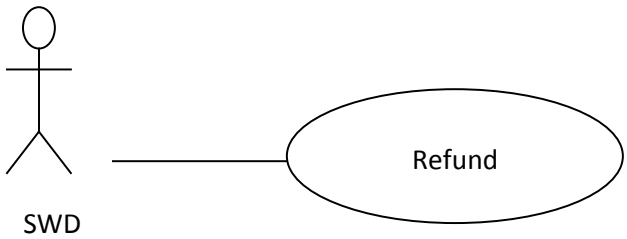
Brief Description: SWD makes the payment from the user's account

Initial Step-by-step Description:

1. Once the cab is confirmed, SWD makes the transactions from the student's account.

Use case: Refund

Diagram:



Brief Description: SWD refunds amount if cab is cancelled

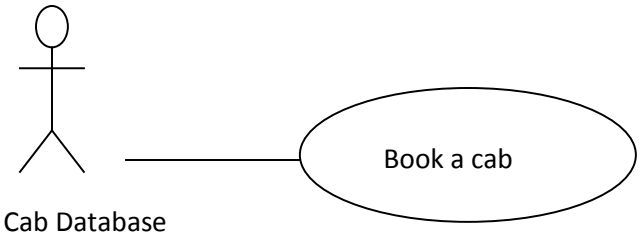
Initial Step-by-step Description:

1. If in case the passenger cancels the cab, part of the amount is refunded via SWD.

Cab Database Use Cases

Use case: Reserve cab

Diagram:



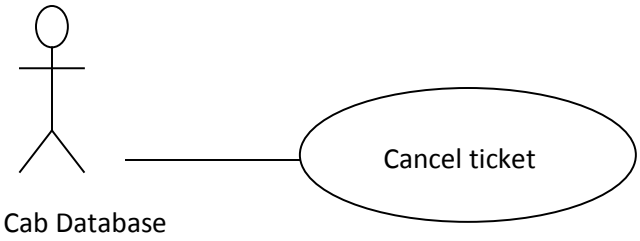
Brief Description: Database updates itself on the journey/passenger/cab details

Initial Step-by-step Description:

1. Once a cab is confirmed for booking only (not sharing), the db updates the cab's availability and removes it from the available cabs list.
2. Once a cab is confirmed for sharing, the db collects all the information about the passengers, groups the data and sorts it according to date/time/route.
3. The passenger is then updated about the cab/co-passenger details.

Use case: Cancel ticket

Diagram:



Brief Description: Database cancels the ticket if the time of cancellation is within the accepted time-limit

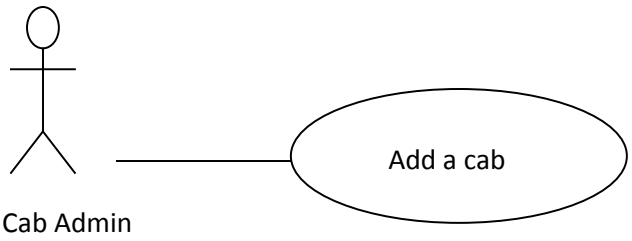
Initial Step-by-step Description:

1. If in case the passenger decides to cancel the cab, the db checks for time constraints and then cancels the cab if possible, and removes the passenger from its list.

Admin Use Cases

Use Case: Add a cab

Diagram:



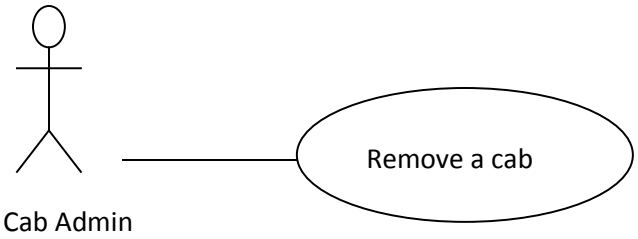
Brief Description: An extra cab is added to the database

Initial Step-by-step Description:

1. Admin has special privileges to add a cab to the db.

Use Case: Remove a cab

Diagram:



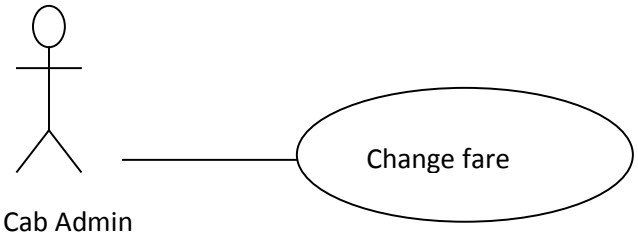
Brief Description: A cab is deleted from the database

Initial Step-by-step Description:

1. Admin has special privileges to remove a cab from the db.

Use Case: Change fare

Diagram:



Brief Description: Admin can change fare for any route in the db

Initial Step-by-step Description:

1. Admin has special privileges to change the fare when necessary

3.3 User Characteristics

The student is assumed to be valid student of BITS Goa with sufficient balance in the SWD Account. In case of insufficient balance, the charges are considered to be a part of semester dues.

Admin logs in like a general user with a username and password, and the system then identifies the admin and grants special privileges.

The people's database is considered to have a complete list of students, faculty members, ARC, AVU, PRO, etc.
Room database is assumed to have the complete list of all rooms on campus.
Cab list is a flexible list available to the admin/service provider who can regularly update cab details.

It is assumed also that SWD is integrated with the software for convenient money transactions.

3.4 Non-Functional Requirements

The Resource Management Portal will be on a server with high speed Internet capability. The software developed here assumes the use of a tool such as Tomcat for connection between the Web pages and the database.

2.0 Summary

This project aims to maximize efficiency and convenience for both students as well as administrators for better handling of campus resources. This in turn minimizes any manual work involved and helps save time and increase accuracy in the various processes involved.