

Second Draft

Travelling and Daily Allowance Management System

Software Requirements Specification



Submitted By

Nikhil Srivastava (B16CS020)

Zaid Khan(B16CS040)

Second Draft

Revision History			
Date	Version	Description	People
18-02-2018	1.0	First draft	Nikhil and Zaid
11-03-2018	2.0	Second Draft	Nikhil And Zaid

Table of Contents

1. Introduction	
1.1 Purpose	4
1.2 Scope	Error! Bookmark not defined.
1.3 Constraints	4
1.4 Assumptions and Dependencies	Error! Bookmark not defined.
1.5 Definitions, Acronyms and Abbreviations	5
1.6 References	5
1.7 Organisation of Documentation	5
2. Overall Description	5
2.1 Functional Requirements	6
2.2 Use Case Diagrams	6
2.3 Class Diagram	6
2.4 Sequential Diagram	8
2.5 Activity Diagram	11
3. Specific Requirements	
3.1 Use Case Description	13
3.2 Supportability	17

Second Draft

Software Requirements Specification

1. Introduction

1.1 Purpose :

This document serves the purpose of illustrating the requirements of the software (functional as well as non-functional) –Travel Allowance Management System. This software makes the process of reimbursement hassle free and efficient.

1.2 Scope :

This software deals with the management of the travel allowances borne by the students, professors and the college staff.By the means of this software, the user would be able to apply for the reimbursement of travel allowance, after college funded trips. This software will create an interface, where the user can register, login, using respective credentials and apply for allowances. These allowance requests, will then be processed by the concerned authorities (by means of this software).

1.3 Constraints

This software fails to handle concurrent requests to access data.It is not possible to integrate this software , with a pre-existing software.Choice of database (text files) makes the process of information editing and retrieval, cumbersome in terms of both space and time.

1.4 Assumptions and Dependencies

1. The users and admin are part of this college itself.
2. The maximum allowance, for which a user can apply for is known apriori to the user.
3. Each user will be approved by a faculty mentor, who will be known to the student apriori.
4. This faculty mentor would be having the power to approve or deny the reimbursement request.
5. The database, where information about the users is stored, is well secured from hackers.
6. The end-user should have a working knowledge of this software.

1.5 Definitions, Acronyms and Abbreviations

Term	Definition
Users	The party which uses the program that includes students, professors and staff of a particular college.
Admin	Administrative authority of the institute
Reimbursement	The process of requesting for allowance money, by submitting the required details.
Mentor	The mentor maybe a student or a professor, who will

Second Draft

	approve the reimbursement request.
PG	Post Graduate
UG	Undergraduate students

1.6 References

[1] www.slideshare.net

[2] Roger S. Pressman, Software Engineering: A Practitioner's Approach

1.7 Organisation of Documentation

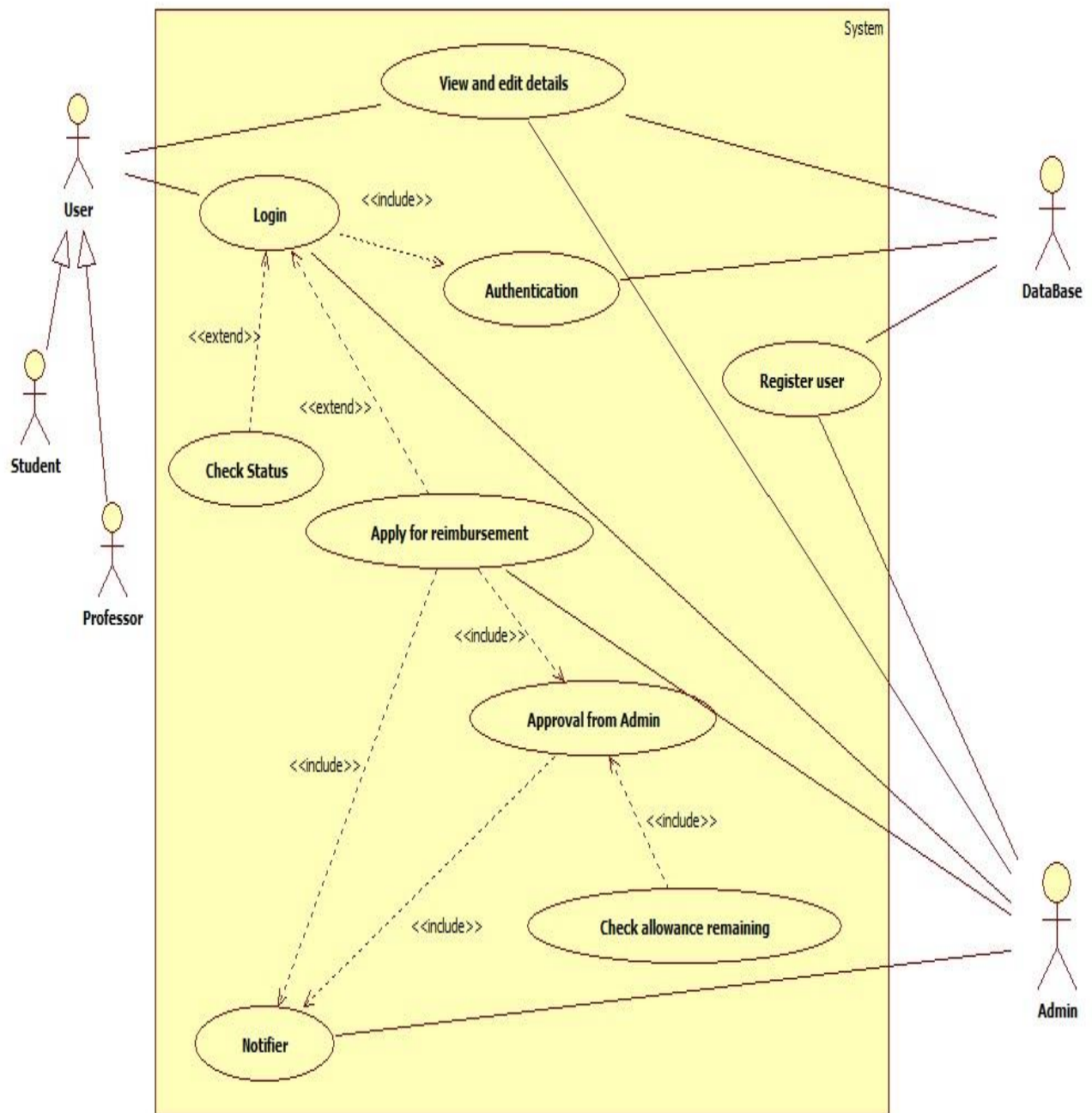
The first section of this documentation involves functional requirements or the capabilities of the software and the available functionalities for the user. It involves proper class diagrams, use-cases with description, sequential diagrams and activity diagrams. The next section describes the various possible non functional requirements including system and software requirements for the proper functioning of the software.

2. Functional Requirements

A new user can register using his credentials. Similar registration routines for students and professors. The user can apply for travelling allowances and the daily allowances borne by him during the journey, in two phases. Professor can get the allowance before as well as after the trip. Proper trip details will be fetched from the user and will then be sent to the mentor (of that trip) for authorisation and further verifications. Upon the approval from the admin, the allowance will be approved and will concurrently be updated in the database.

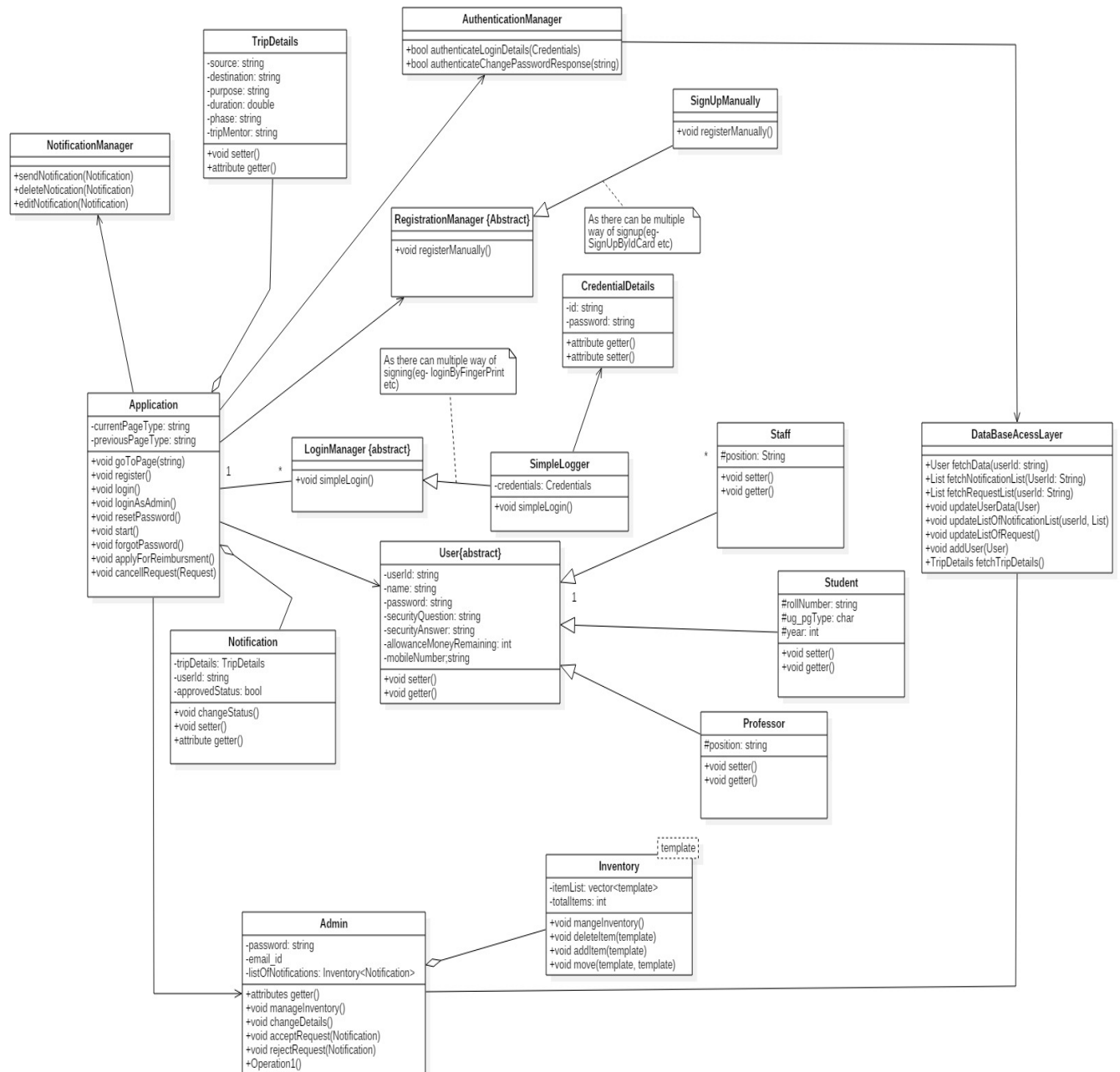
Second Draft

1. Use Case Diagram



Second Draft

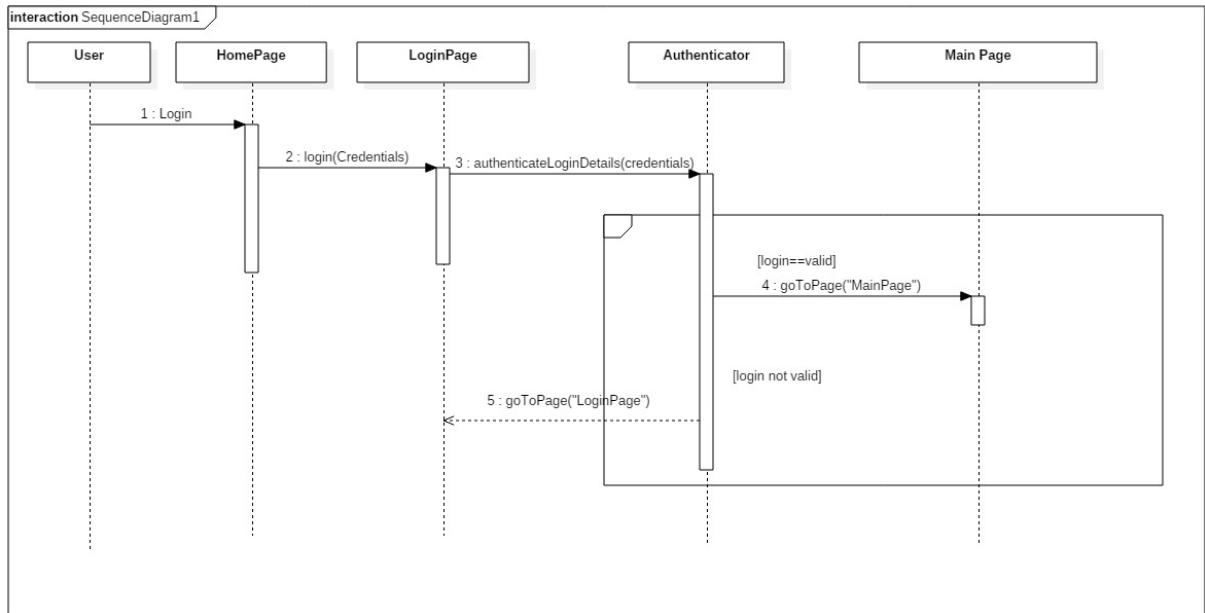
2. Class Diagram :



Second Draft

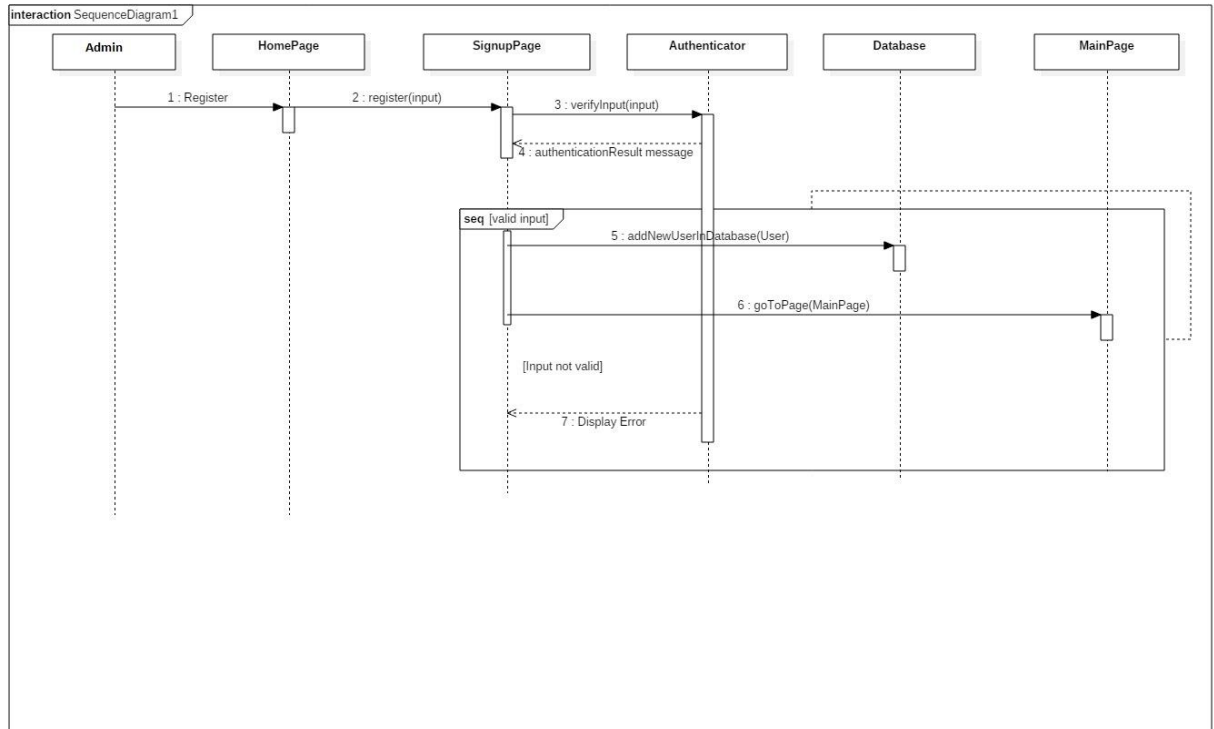
3.Sequential Diagram :

1. Login



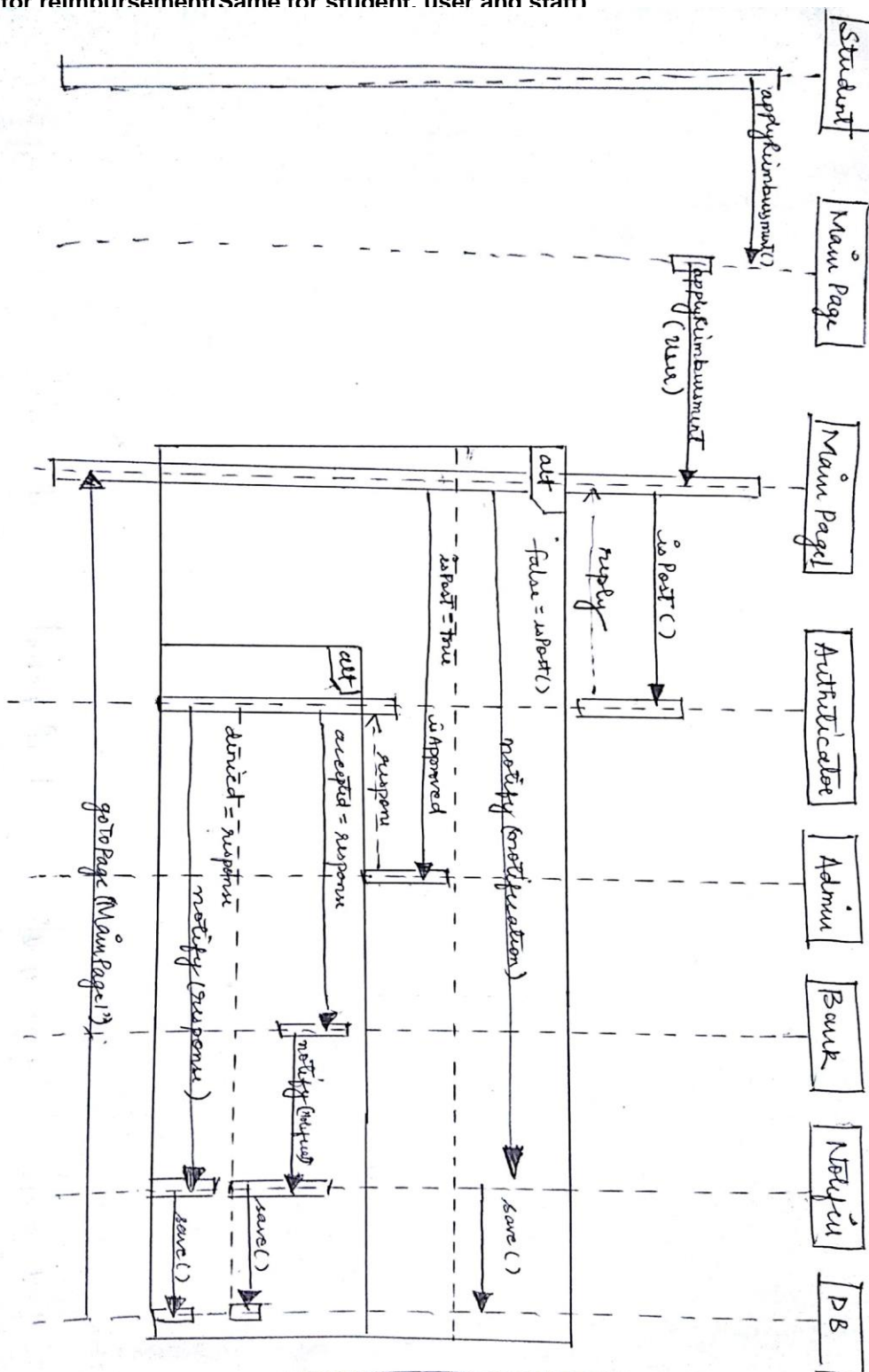
Second Draft

2 Register



Second Draft

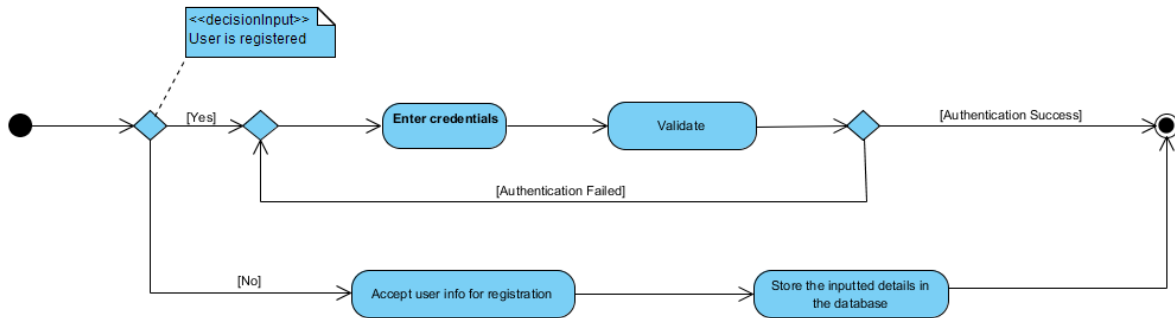
3 Applying for reimbursement(Same for student, user and staff)



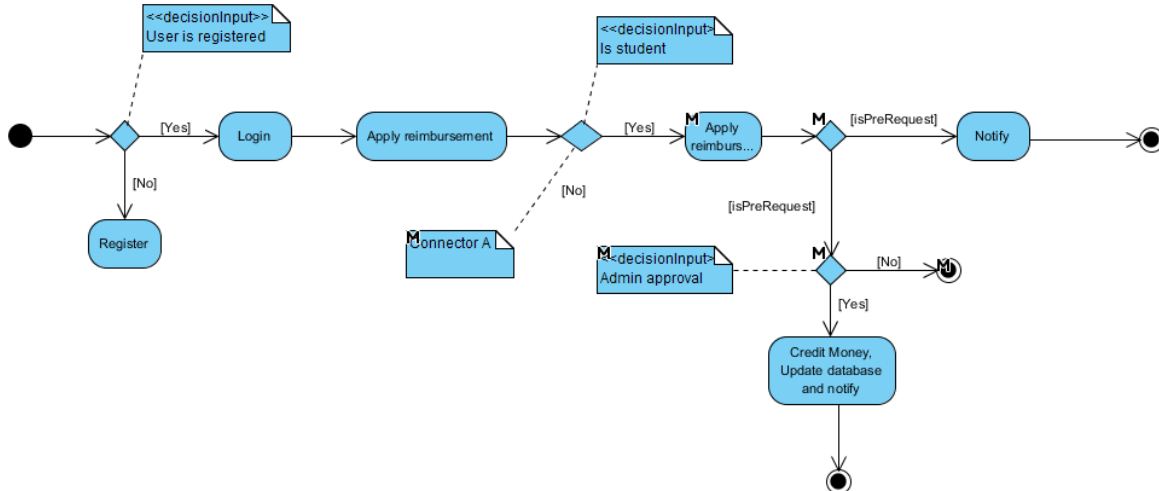
Second Draft

4. Activity Diagrams :

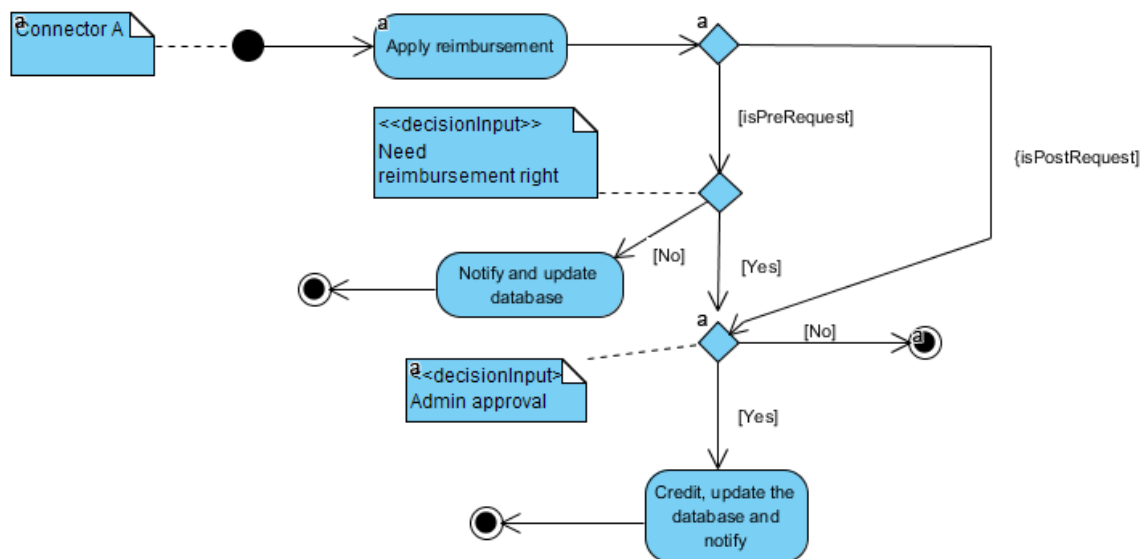
1 Validating credentials



2 Applying for reimbursement



Second Draft



3. Non-Functional Requirements

3.1 Performance Requirements: The response time of the program depends on the connectivity with the database, (local host in this case). So response time is less than 1 second.

3.2 Security Requirements : User data is well secure in the database and can't be accessed without successfully logging into the system.

3.3 System Requirements : The system must be able to run full -fledged C++ applications, with pre-installed GCC compiler.

3.4 Software Requirements : The system should be pre-installed with libraries such as windows.h and conio.h, to allow uninterrupted working.

4.User Characteristics

To be precise there are 3 types of users who will interact with this software. Namely, Students (UG,PG) , Professors and Admin.

Students and Professors will have powers to register, change their details and apply for allowances . Request for allowance can be applied for, both before and after the trip. They can even access their previous allowances for which they have applied for in the past.

The Admin, will have the power to authorise or deny any allowance request. He/She will have a proper access to the database, in case he wants to review some past claim.

Second Draft

Use case description

Name	Login
Short Description	The user or admin, would feed their valid user-id and password, to log into the software.
Precondition	Entered user-id and password are valid and exist in the database.
Postcondition	User is successfully logged into the software.
Error Situations	Invalid credentials or file not accessible.
System State in the state of error	Error is printed.
Actors	User (including students and professors) and Admin
Trigger	After login option is invoked from the main interface page.
Standard Process	<ol style="list-style-type: none">1. User enters the credentials.2. Details get authorised from the database.3. In case of successful login, user is redirected to menu page else error message is printed and redirected to main interface page.
Alternative Process	User can get work done manually.

Name	View and Edit Details
Short Description	The user will be able to view and change his/her respective account details. Here, Admin will have the exclusive power to access and change details of any arbitrary user.
Precondition	User should be an existing member of this software and should be able to login using his credentials.
Postcondition	Information will be fetched from the database, and can be accessed by the user
Error Situations	In case of wrong inputs.
System State in the state of an error	Error message is shown.
Actors	User (student and professor) , Admin , Database.

Second Draft

Trigger	Once view and edit details option is invoked from the main page
Standard Process	1.In case user invokes the view details option: <ul style="list-style-type: none">• User details are fetched from the database and stored in a temporary user object.• Then details are displayed across the application interface. 2.In case user invokes edit details option : <ul style="list-style-type: none">• The software updates the details as edited by the user, in the database.
Alternative Process	As an alternate, the user may directly go the concerned authority's office to view and edit details.

Name	Register
Short Description	A new student or professor, can register his/her details and become a user of the software.
Precondition	The new member must not be an existing user of the software.
Postcondition	Details of the new member are updated in the database and he/she becomes a user of the software and he/she will be redirected to the main menu page.
Error Situations	In case the new member is a pre-existing user, an error message will be displayed. Or in case a mismatching data is entered in some data field.
Actors	Data Verifier, Database and User
Trigger	Upon invoking the register option from the main interface menu.
Standard process	1. User feeds his/her required details. 2.Program ensures, that the entered data type matches the required type. 3.After checking the error situations, database is updated and user is redirected to the main menu page.
Alternative Process	The new person may visit the concerned authority's office and get registered.

Name	Authentication
Short Description	Matches the data entered by the user, with the database. If matched, redirects to the main menu page. Else, displays error message.

Second Draft

	If user forgets password, then upon verification of the security question from database, new password may be created.
Precondition	User must have entered his/her credentials in the login page.
Postcondition	Data gets available for fetching from the database.
Error Situation	<ul style="list-style-type: none">• If user does not exist, error message is displayed and asking to register.• If password does not match, error message is displayed.
Actors	Data Verifier
Trigger	After entering the user details, authentication is maneuvered.
Standard Process	<ul style="list-style-type: none">• Verifies the entered data with the database.• If verified successfully, data becomes accessible.• Else error message is displayed.• If user forgets password, then upon verification of the security question from database, new password may be created.
Alternative Process	User can directly mail his/her problem to the concerned authority (Admin).

Name	Apply for reimbursement
Short Description	This maneuvers the whole process of applying for the reimbursement. Processes the request and sends it to the admin for final approval.
Precondition	The user must be having a proper trip detail proof, in order to claim the reimbursement.
Postcondition	If the claim gets approval from the admin, then the allowance is given and successful notification is sent to the user, else error notification is given.
Error Situation	In case balance allowance is not cleared, and still the user applies for yet another allowance.
System State in the state of an error	Error message is shown.

Second Draft

Actors	Admin
Triggers	After entering the trip details and then invoking, applying for reimbursement.
Standard Process	<p><u>Applying for allowance before the trip:</u></p> <ul style="list-style-type: none">• User must ensure that the previous unpaid allowances are completed.• Logged in professor, fills in the trip details and send the request for allowance to the admin.• The admin, processes the request and takes decision.• Upon approval allowance is given and concurrently database is updated.• After returning from the trip, it is required, by the user to submit proper proof of trip details and return the balance amount to the concerned authority <p><u>Applying for allowance after the trip :</u></p> <ul style="list-style-type: none">• The user must ensure that the previous unpaid allowances are completed.• After this the user, enter the trip details, which are sent to the admin for approval, admin processes the request.• Upon approval from the admin, allowance is given and database is updated.
Alternative process	User may fill in the hardcopy of forms manually and submit it to the concerned authority.

Name	Notifier
Short Description	This use case sends notification in case a user applies for an allowance, gets approved or gets denied (in all three cases a separate notification is sent to the user and to the admin.)
Precondition	The user should have applied for an allowance.
Postcondition	Incase the user has just applied for the allowance, the Admin will get a notification regarding the same. Incase the allowance gets approved by the admin, a successful notification will be sent. And same for the denial case.
Error Situation	Wrong inputs given
System State in the state of an error	Error message is displayed.

Second Draft

Actors	Admin
Trigger	Applying for the allowance triggers the process of sending the notifications.
Standard Process	By checking the status of the approval, a notification is updated in the database for the users and admin.
Alternative process	User has to contact office personally in order to check status .

Maintenance:

The program is highly maintainable as, in the future if need arises to integrate our software with any other software, it can easily be done because of the diverse nature of the classes. All aspects are considered regarding maintainability. In future, if way of registration or login is modified, then it can easily be integrated, thus making it highly maintainable.

Design is language agnostic, so it can be implemented by ANY language.

Supportabilty:

Naming Convention : The code will be written as specified by the Hungarian Naming Convention.