Software Requirements Specification

Version 1.0
<<Annotated Version>>

October 29, 2014

Mess Management System

Abhinav Srivastava
Varun V Gopal
Anuj Modi
Siddhant Pardeshi
Tejas Jogi
Suhail Mannan Jadliwala
Sagar Verma

Submitted in partial fulfillment
of the requirements of
CS F213
Object Oriented Programming

<<Any comments inside double brackets such as these are *not* part of this SRS but are comments upon this SRS example to help the reader understand the point being made.

This work is based upon the submissions of the Project 2014 CS F213. The students who submitted these team projects were Abhinav Srivastava, Varun V Gopal, Anuj Modi, Siddhant Pardeshi, Tejas Jogi, Suhail Mannan Jadliwala, Sagar Verma.
. >>

Table of Contents

1.0 Introduction

1.1. Purpose

The purpose of this document is to present a detailed description of the Mess 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 users and the developers of the system.

1.2. Scope of Project

This software system will be a Mess Management System for the Students and Mess Staff. This system will be designed to ease the work of choosing the mess, viewing the menu, applying for leave for students. And from admin's point of view it will help them to keep better track of the huge database of the people enrolled in the mess, displaying the menu and keeping track of the employees and food stock and their payments.

1.3. Glossary

Term	Definition
Guest	Any person who has not paid a Mess Advance
Database	Collection of all the information monitored by this system.
Mess Crew	The Mess Staff: Cooking, Serving and Cleanup
Mess Admin	The person who has access to the database and manages the server
Student	A person who is studying in the college and has paid a Mess Advance
Software Requirements Specification	A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document.
Server	The central terminal that handles authentication and databases.
User	Anyone who uses the system

1.4. References

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

1.5. Overview of Document

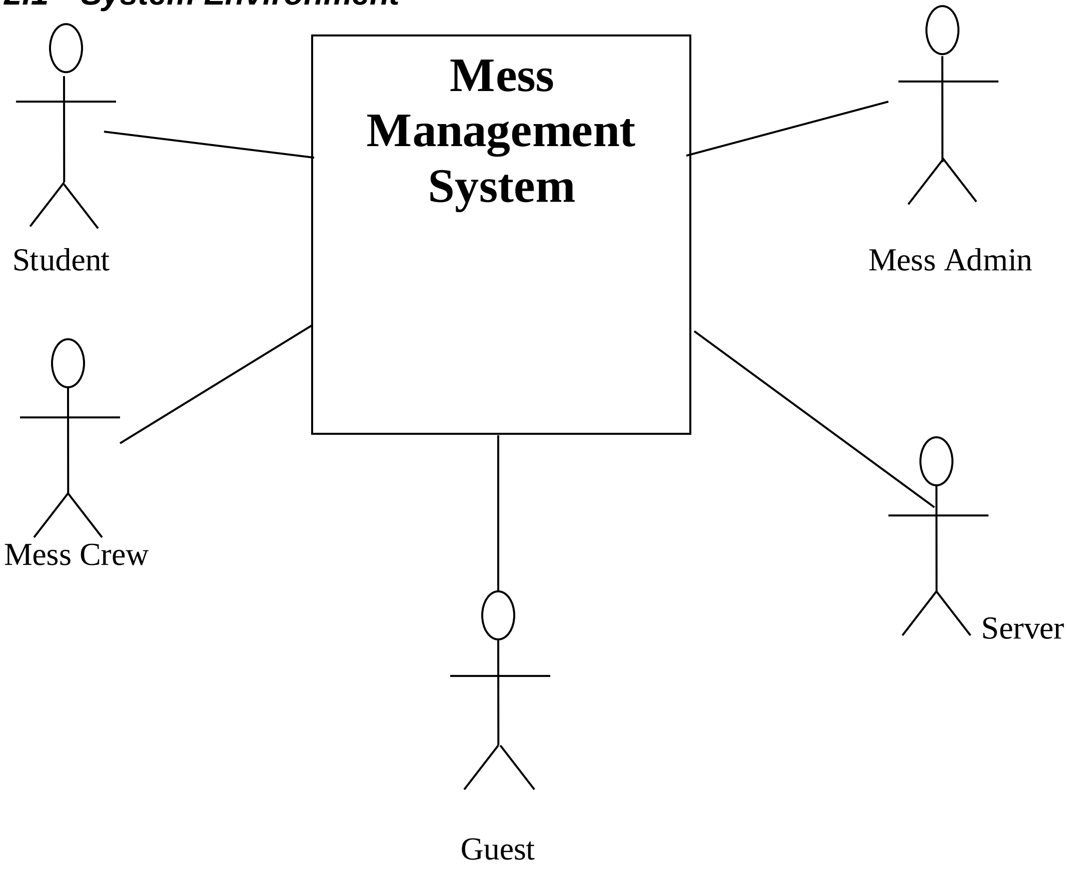
The next chapter, the Overall Description section, of this document gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter.

The third chapter, Requirements Specification section, of this document is written primarily for the developers and describes in technical terms the details of the functionality of the product.

Both sections of the document describe the same software product in its entirety, but are intended for different audiences and thus use different language.

2.0 Overall Description

2.1 System Environment



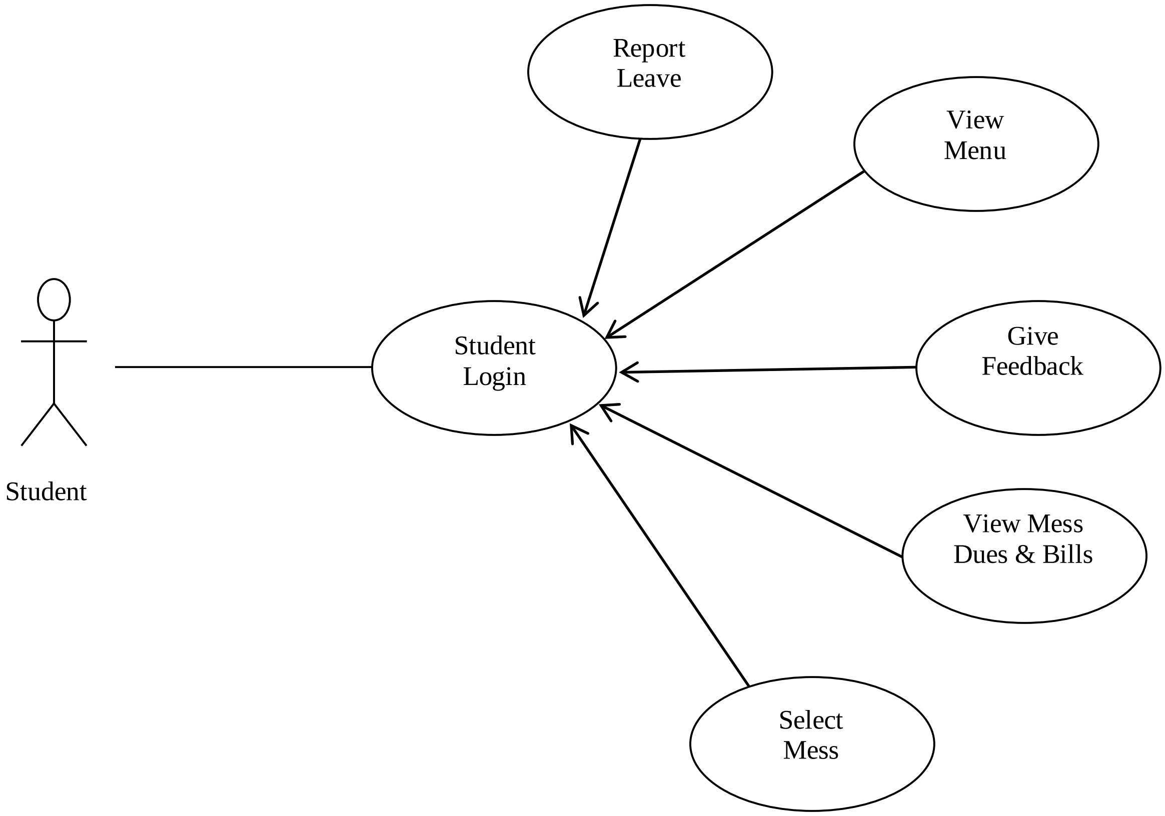
The Mess Management System has five active actors. Student, Guest, Mess Crew, Mess Admin and Server.

2.2 Functional Requirements Specification

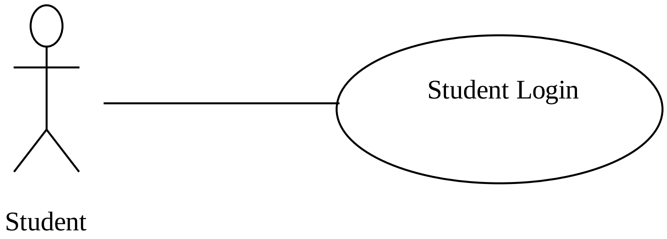
This section outlines the use cases for each of the active actors separately. The Student, the Mess Crew, the Mess Admin, the Guest and Server.

2.2.1 Student

Student Use Case



Use case: Student Login
Diagram:



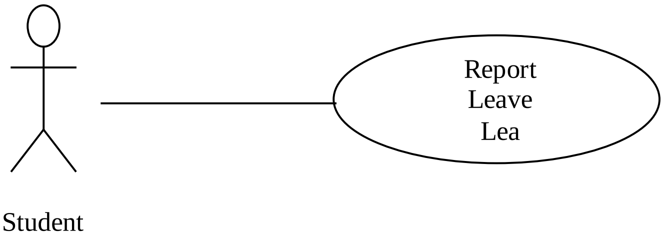
Brief Description

The Student is authenticated in to the Mess Management System using their ID Number.

Initial Step-By-Step Description

1. The Student fills in their ID number either by typing or scanning the bar code.
2. The System validates their ID.
3. If login is valid, the system displays a menu with options for student

Use case: Report Leave
Diagram:

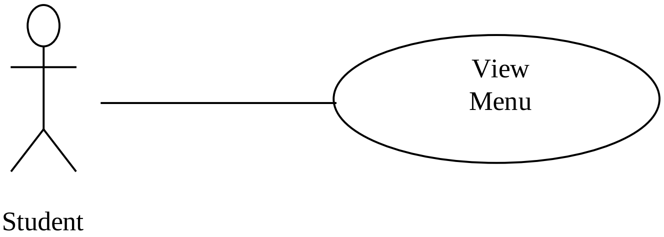


Brief Description
The Student reports absence from college to the mess, to avail concessions.

Initial Step-By-Step Description
Before this use case can be initiated, the Student has already logged in

1. The Student logs in and selects the Report Leave option.
2. The Student specifies the date and time of arrival and departure.
3. The System stores the information in the server and uses it accordingly to manage the due details of the Student.

Use case: View Menu
Diagram:



Brief Description
The Student views the menu in either mess for the next week.

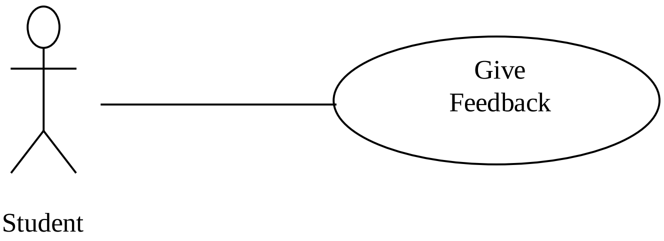
Initial Step-By-Step Description

Before this use case can be initiated, the Student has already logged in

1. The Student logs in and selects the View Menu option.
2. The Student selects the day of the week and the meal (i.e. Breakfast, Lunch, Evening Snacks, or Dinner) for which to view the menu. The student also has the option to view the menu for the current meal time.
3. The menu for the requested day and time is displayed depending on the Student's opted mess by the System. If the student selected current meal time, the system synchronizes the time with the server and fetches the details accordingly.

Use case: Give Feedback

Diagram:



Brief Description

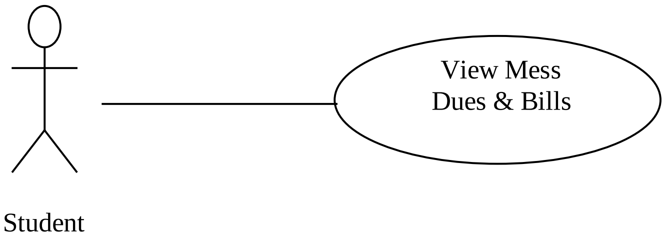
The Student can give feedback regarding the quality of food, cleanliness of mess, etc.

Initial Step-By-Step Description

Before this use case can be initiated, the Student has already logged in

1. The Student logs in and selects the Give Feedback option.
2. The Student specifies the type of feedback/complaint by selecting from a list provided.
3. The Student gives a brief description in the area provided, and submits the form for reference.
4. The system stores the feedback in the server to make it viewable to the Mess Admin.

Use case: View Mess Dues & Bills
Diagram:

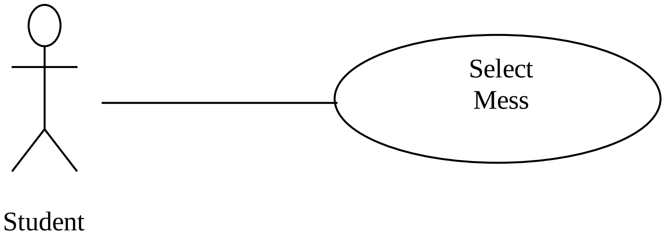


Brief Description
The Student views the Mess Dues and summary of already paid bills.

Initial Step-By-Step Description
Before this use case can be initiated, the Student has already logged in

1. The Student logs in and selects the Mess Dues & Bills option.
2. The Student selects the month for which to view Dues or Bills.
3. The System retrieves the requested information from the server database and makes it available to the student.

Use case: Select Mess
Diagram:



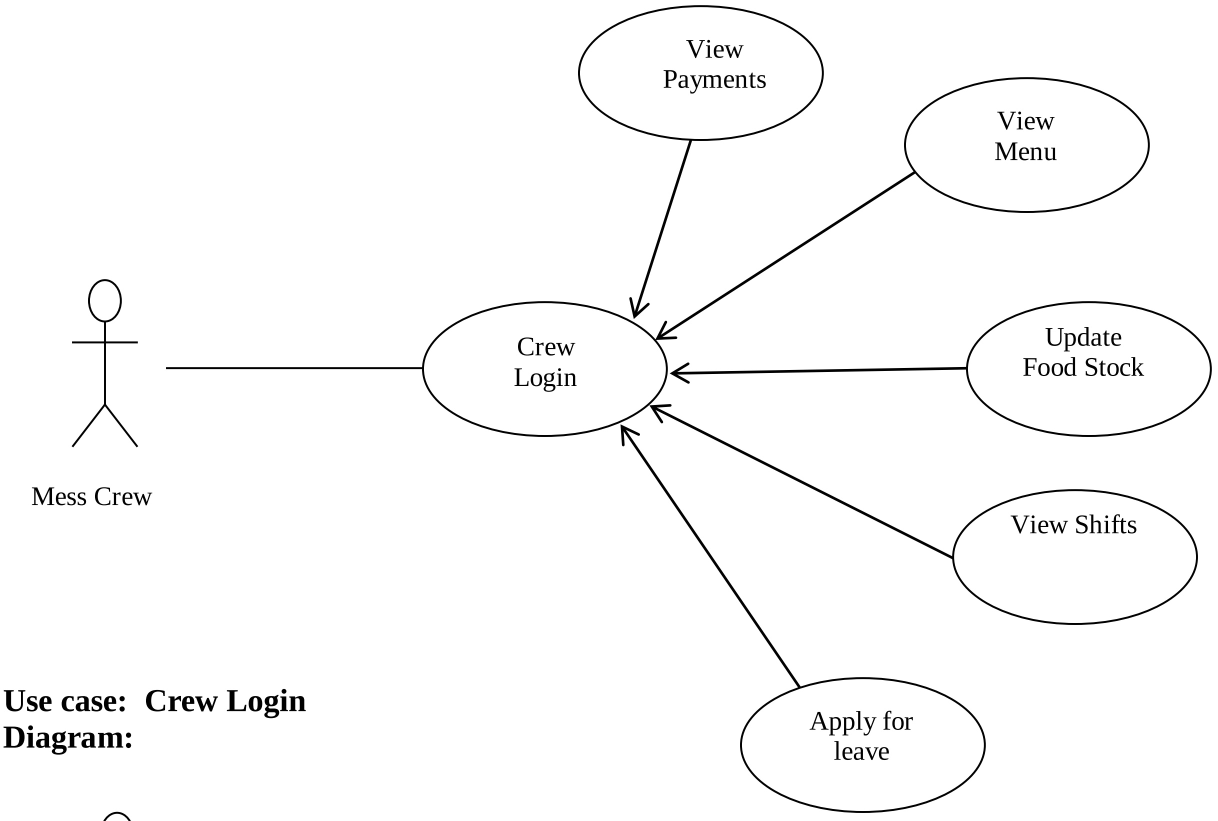
Brief Description
The Student selects the mess for the next month.

Initial Step-By-Step Description
Before this use case can be initiated, the Student has already logged in

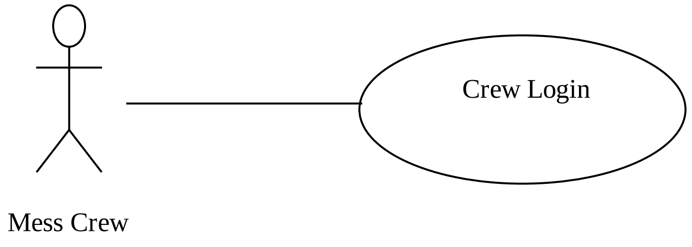
1. The Student logs in and selects the Select Mess option.
2. The System provides the Student a choice between A & C mess provided the deadline for choosing hasn't yet passed. If the deadline has passed, this option will be greyed.
3. The System stores the choice and updates the Student Details with the server accordingly.

2.2.2 Mess Crew

Mess crew Use Cases



Use case: Crew Login
Diagram:

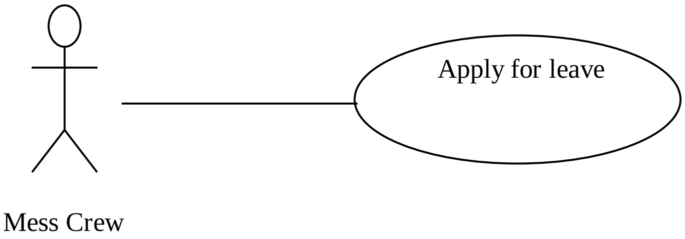


Brief Description
The Crew logs in to the Mess Management System using their username and password.

Initial Step-By-Step Description

1. The Mess Crew member fills in their username.
2. The member enters their unique password.
3. The System validates their login.
4. If login is valid, the system takes the person to their homepage

Use case: Apply for leave
Diagram:

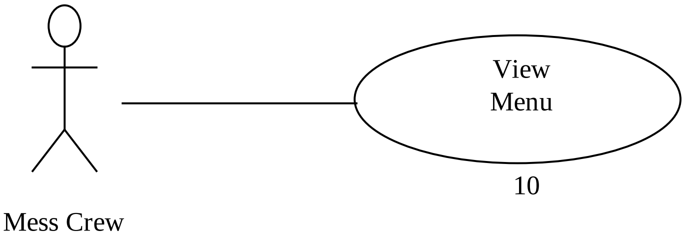


Brief Description
The Crew member requests the admin for leave.

Initial Step-By-Step Description
Before this use case can be initiated, the Crew member has already logged in.

1. The Crew member logs in and selects the Report Leave option.
2. He/she specifies the date and time of arrival and departure.
3. The System forwards the application to the mess admin and updates the database.
4. The Mess Admin has the ability to approve or disapprove the leave request and thus manage crew shifts.
5. The shift database is updated according to the mess admin's decision.

Use case: View Menu
Diagram:

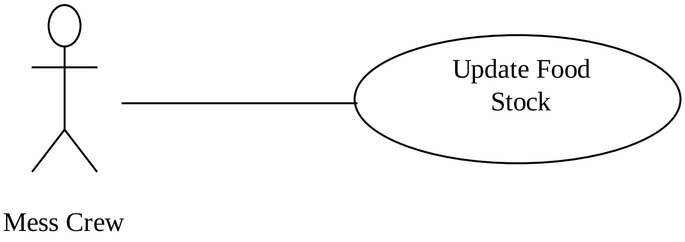


Brief Description
The Mess Crew member views the menu in mess for the next week.

Initial Step-By-Step Description

Before this use case can be initiated, the Mess Crew has already logged in
1. The Mess Crew member logs in and selects the View Menu option.
2. He/she selects the day of the week and the meal(i.e. Breakfast, Lunch, Evening Snacks, or Dinner)for which to view the menu.
3. The menu for the requested day and time is displayed.

Use case: Update Food Stock
Diagram:

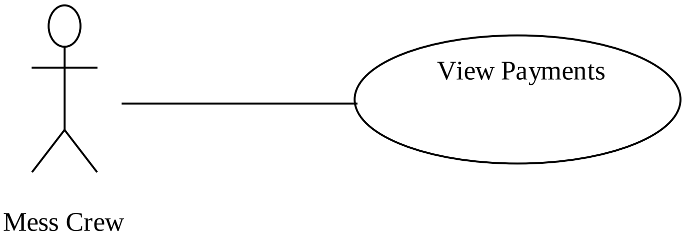


Brief Description
The Mess Crew can update the information regarding the Food Stock.

Initial Step-By-Step Description
Before this use case can be initiated, the Mess Crew has already logged in

1. The Mess Crew logs in and selects the Update Food Stock option.
2. The Mess Crew updates the current stock info.
3. The system stores the updated info to the database and the server.

Use case: View Payments
Diagram:



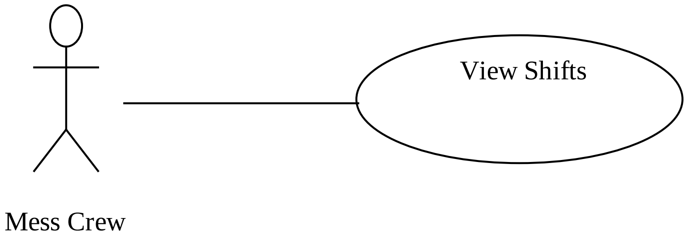
Brief Description
The Mess Crew views the details of their pending payments.

Initial Step-By-Step Description
Before this use case can be initiated, the Mess Crew has already logged in
1. The Mess Crew member logs in and selects the View Payments option.

2. He/she selects the month for which to view Payments.

3. The System retrieves the requested information from the server's database and makes it available to him/her.

Use case: View Shifts
Diagram:



Brief Description

The Mess Crew views the information regarding shifts of work.

Initial Step-By-Step Description

Before this use case can be initiated, the Mess Crew has already logged in

1. The Mess Crew logs in and selects the View Shifts option.

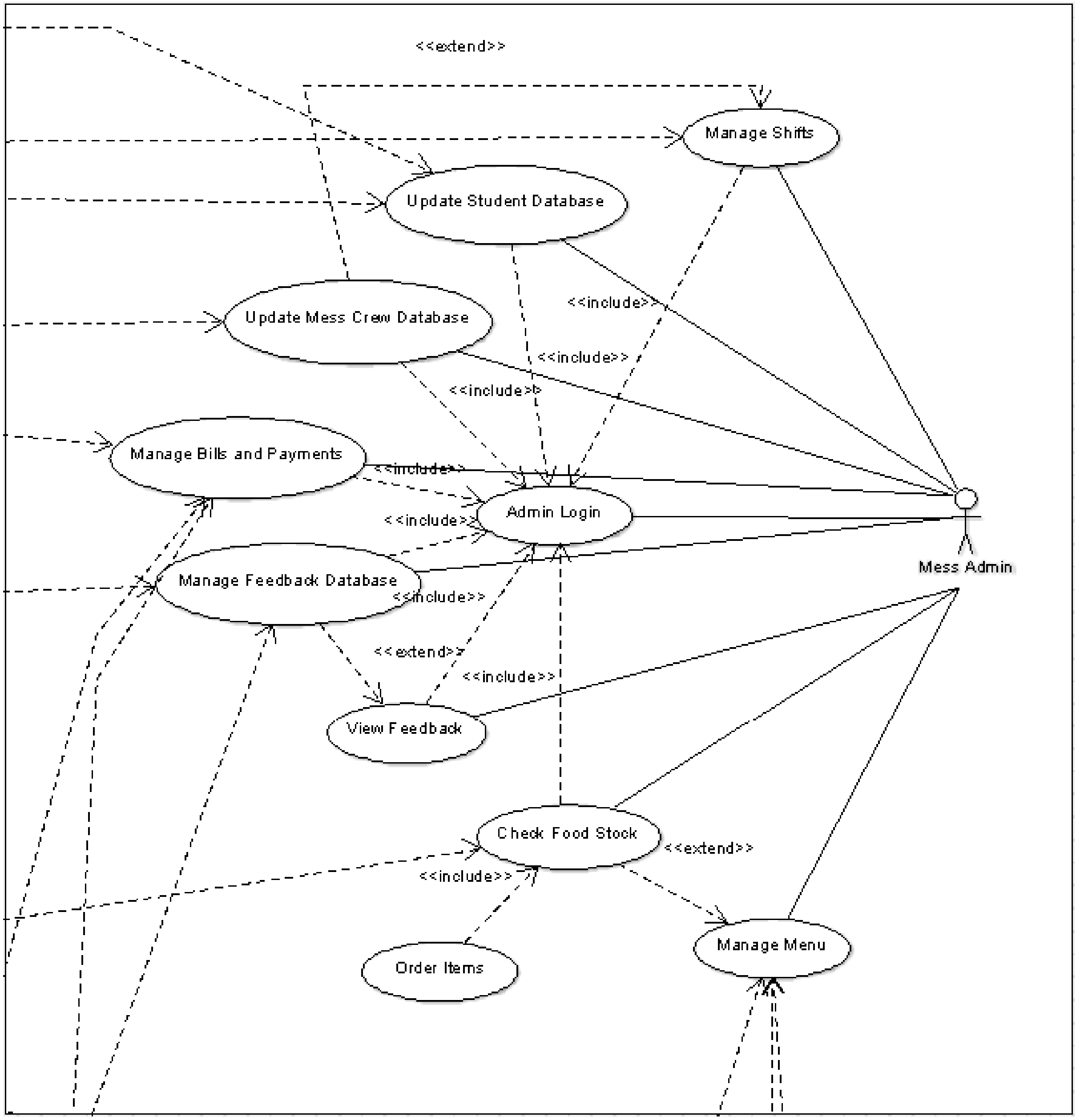
2. The System provides the Mess Crew with an option of the day for which he has to view his shifts.

3. The System fetches the details from the server and displays them accordingly.

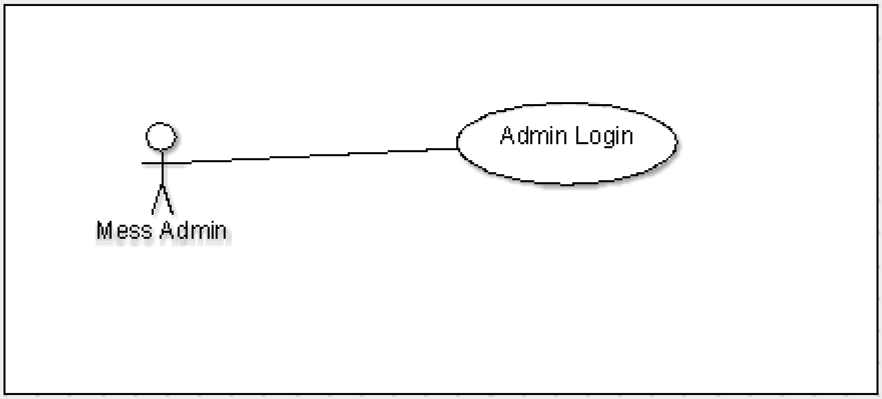
2.2.3 Mess Administrator

Mess Administrator Use Cases

The Mess Admin has the following sets of use cases:



Use Case: Mess Admin Login
Diagram:



Brief Description

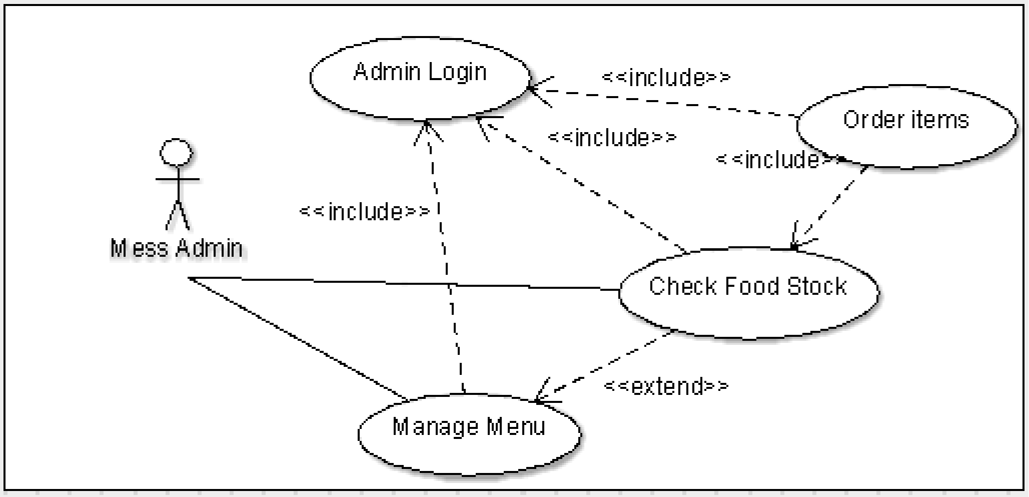
Mess Admin is a person who controls various activities(as mentioned in the use cases diagram) of the mess. The Mess Admin attempts to login and identify himself from the mess terminal in order to access options available to him.

Initial Step-by-Step Description

This use case can be initiated from a dedicated ‘Mess Admin’ portal which is implemented in the software interface by a ‘Mess Admin’ button.

1. The Mess Admin clicks on the ‘Mess Admin’ button
2. The system requests for a username and password to authenticate.
3. The authentication details entered by the Mess Admin are processed by the server.
4. If successful, a menu with options available to admin is displayed.

Use Case: Menu& Stock Management
Diagram:



Brief Description

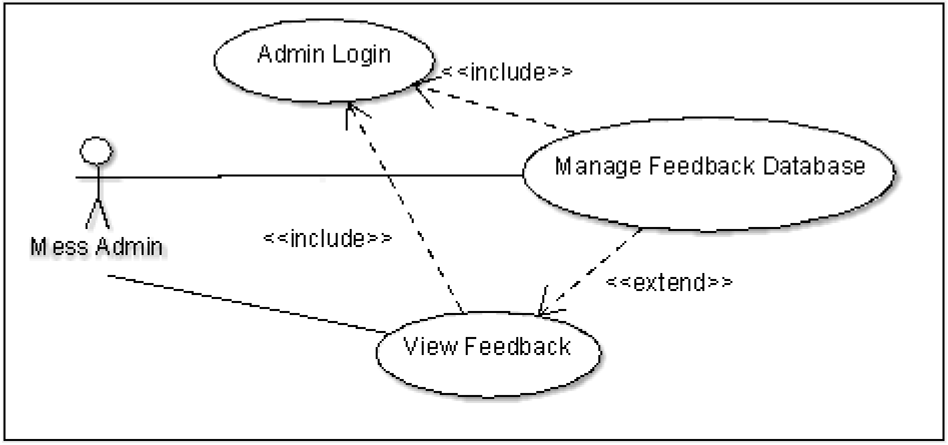
After logging in, the Mess Admin has three options to adjust his menu according to stock available and manage the stock

Initial Description

Initiating this use case requires the Mess Admin to authenticate himself by logging in.

1. The Mess Admin clicks on 'Order Items' to place order for the amount of stock that he needs. For that he needs to check the available stock at "Check food Stock" which the system displays.
2. He can also choose 'Check food Stock' to check stock at his convenience, which the system displays.
3. The Mess Admin can select 'Manage Menu' to change the menu for certain day according food stock available.

Use Case: Check Feedback
Diagram:



Brief Description

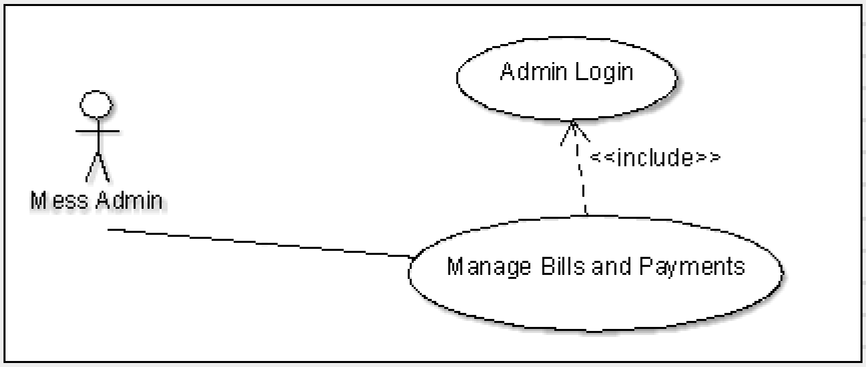
The Mess Admin can check feedbacks from guests and students and manage(add) entries to feedback database manually.

Initial Description

Initiating this use case requires the Mess Admin to authenticate himself by logging in.

1. The Mess Admin clicks on 'View Feedback'.The system presents the admin with all the feedback entries from the feedback database, stored at the server.
2. The Mess Admin clicks on 'Manage Feedback'. He can directly access the database from the server and add or delete entries.

Use Case: Manage Bills and Payments
Diagram:



Brief Description

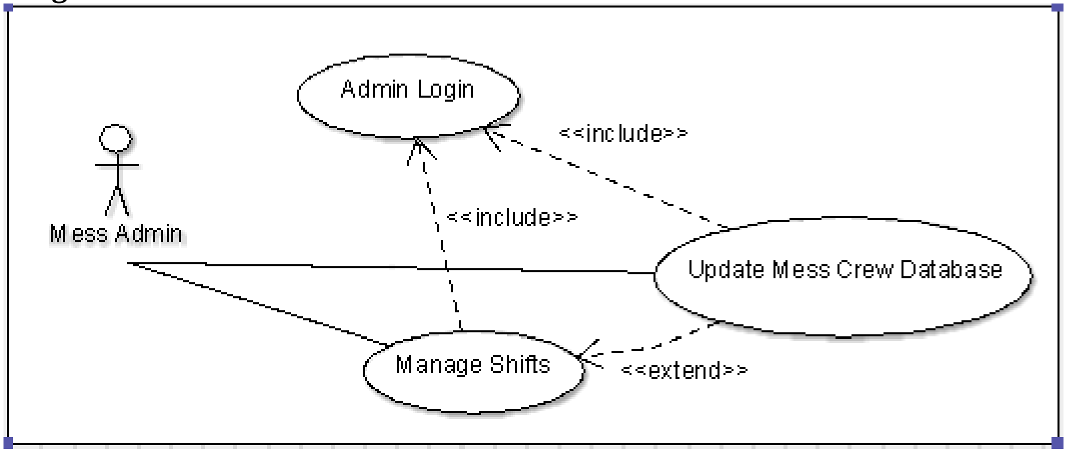
The Mess Admin updates the Payment & Billing database for students, guests and Mess workers to view.

Initial Description

Initiating this use case requires the Mess Admin to authenticate himself by logging in.

1. The Guest clicks on ‘Manage Bills and Payments’.
2. He searches the student or guests who has an outstanding payment or the worker who has to receive salary.
3. He updates their respective profiles (objects) with latest credit or debit to them.
4. The system records these updates and the worker, guests or student can view it when they log in.

Use Case: Crew Management
Diagram:



Brief Description

The Mess Admin can manage the shifts of the mess workers, add new employees and see there leave requests.

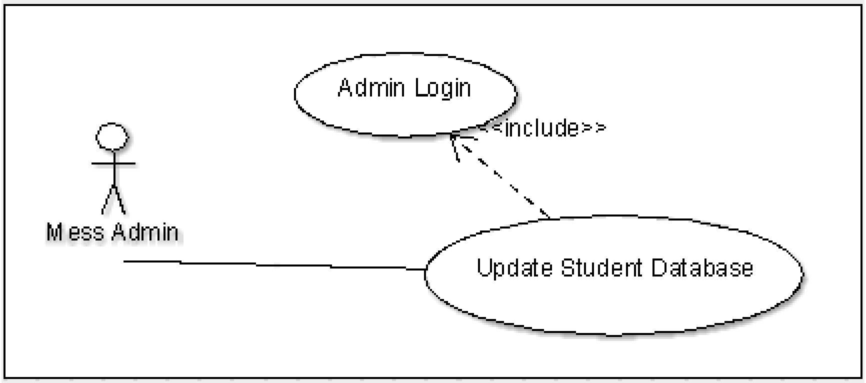
Initial Description

Initiating this use case requires the Mess Admin to authenticate himself by logging in.

1. The Guest clicks on ‘Crew Database’. He has the options to add a new employee and to see worker leave application.

2. The Guest clicks on ‘Manage Shift’. He can change the worker’s shift which the worker can see when he choose ‘View Shift” option. He can determine how to manage these shifts by checking which workers are on leave and who are not from ‘Crew Database’

Use Case: Student Details
Diagram:



Brief Description

The Mess Admin is able to view students enrolled in the mess for that month and view number of students on leave.

Initial Description

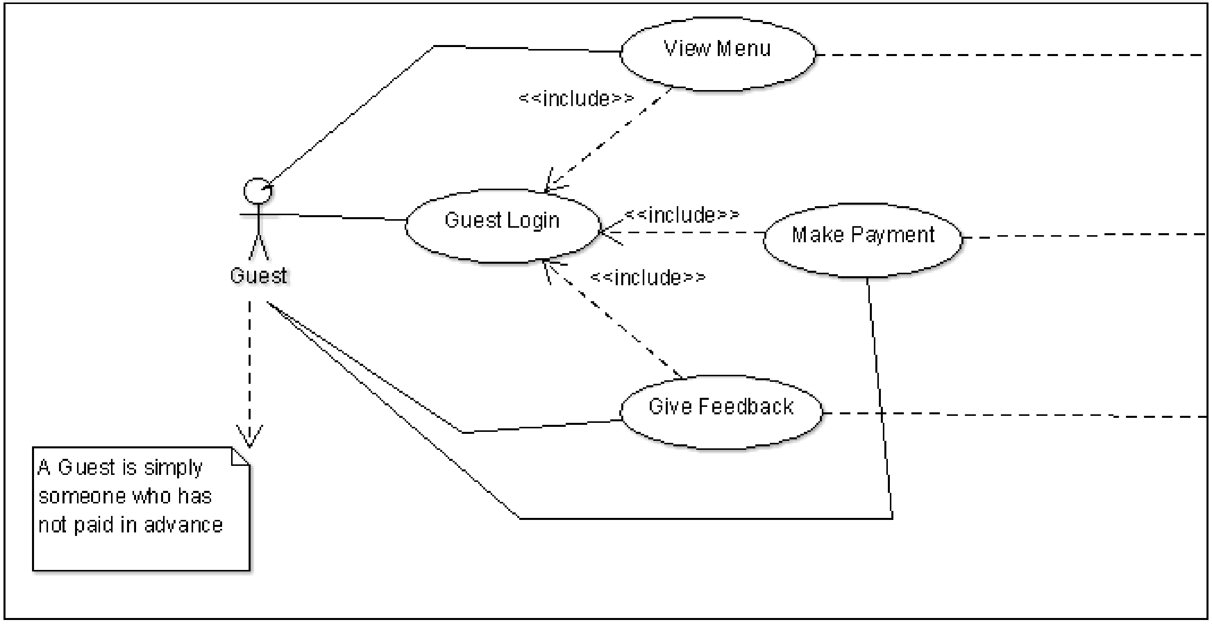
Initiating this use case requires the Mess Admin to authenticate himself by logging in.

1. The Guest clicks on ‘Students’.
2. The system displays all the students enrolled in the mess for that month and alongside displays whether they are on leave that day.

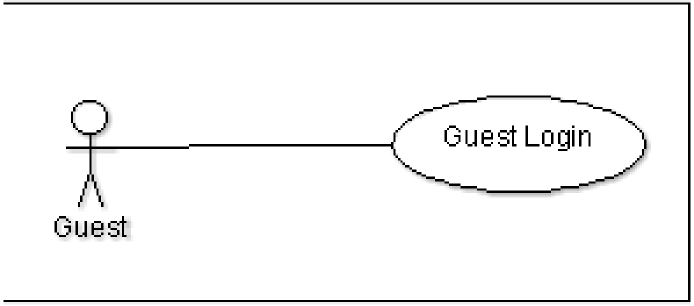
2.2.4 Guest

Guest Use Cases

The Guest has the following sets of use cases:



Use Case: Guest Login
Diagram:



Brief Description

As commented, a guest is a person who has not paid a mess advance. The Guest attempts to login and identify himself from the mess terminal in order to access options available to guests.

Initial Step-by-Step Description

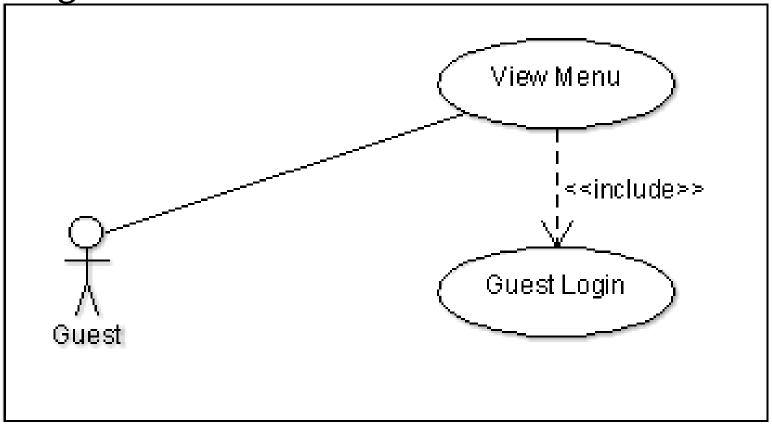
This use case can be initiated from a dedicated 'Guest' portal which is implemented in the software interface by a 'Guest' button.

1. The Guest clicks on the 'Guest' button

2. The system requests for a name to reference, in case the guest leaves feedback.
3. A menu with options available to guests is displayed.

Use Case: View Menu

Diagram:



Brief Description

The Guest views the mess menu for that particular day and time.

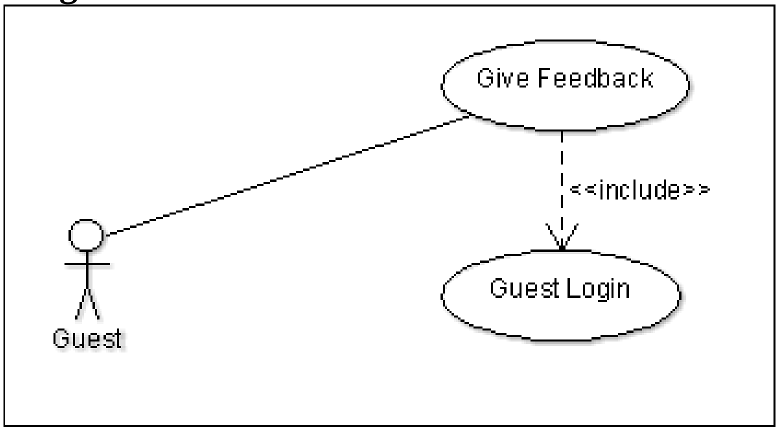
Initial Step-by-Step Description

Initiating this use case requires the guest to identify himself by logging in.

1. The Guest clicks on ‘View Menu’
2. The system presents a list of food items served by the mess on that day at the time of the meal by accessing the Mess Menu database.

Use Case: Give Feedback

Diagram:



Brief Description

The Guest leaves feedback about the mess.

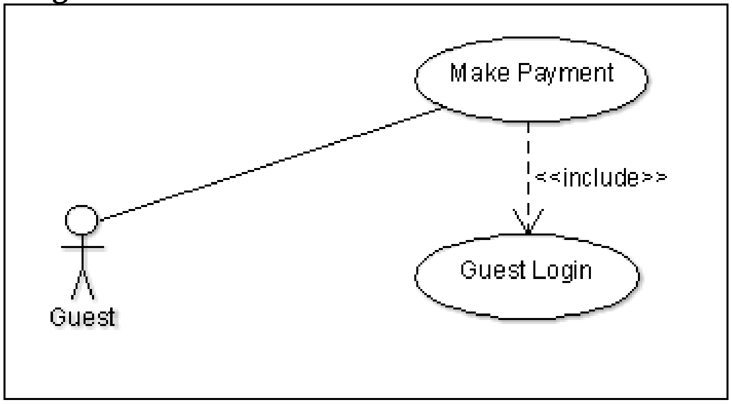
Initial Step-by-Step Description

Initiating this use case requires the guest to identify himself by logging in.

1. The Guest clicks on ‘Give Feedback’
2. The system presents the guest with a form with fields for text input.
3. The Guest enters his input.
4. The system records this feedback by updating the feedback database.

Use Case: Make Payment

Diagram:



Brief Description

The Guest must pay for the meal if he decides to be served.

Initial Step-by-Step Description

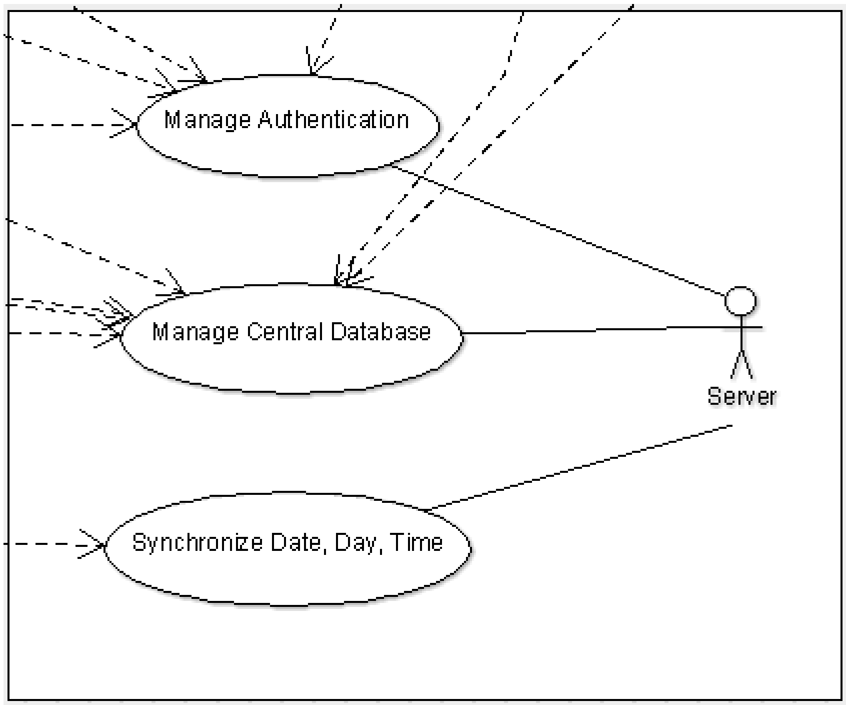
Initiating this use case requires the guest to identify himself by logging in.

1. The Guest clicks on ‘Make Payment’.
2. The system fetches the cost of the meal from the menu and presents it to the guest.
3. The Guest pays cash.
4. The system records the payment as a credit to its accounts.

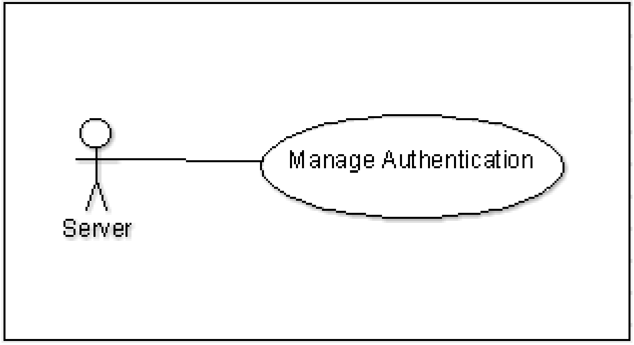
2.2.5 Server

Server Use Cases

The Server actor has been created to manage the Mess Database. This database can be accessed by the Mess Admin only. The Server has the following sets of use cases:



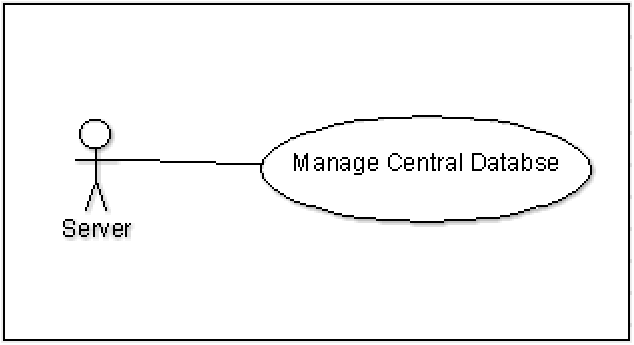
Use Case: Manage Authentication
Diagram:



Brief Description

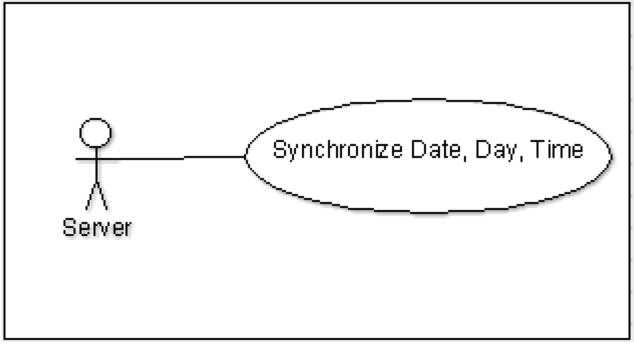
Here, the server manages the authentication of all types of users(Mess Admin, Mess Employee, Student & Guest). For the Mess Admin and the Mess Employee(also Student when he wants to choose mess or enter feedback), the server checks the username and password when one tries to log in. For the Student, the server only checks the ID. For Guest, only his name.

Use Case: Manage Central Database
Diagram:



Brief Description
Here, the server manages changes done to the Student, Crew and Feedback Databases by all the users.

Use Case: Date & Time Synchronization
Diagram:



Brief Description
There are cases when the server provides data, upon request, according to certain dates and time during the day. They are:

1. When any user clicks the ‘View Menu’ option, the server, according to the time & day, displays the menu of the forthcoming meal (at 11:00am on the Monday, View Menu will show Monday Lunch Menu).
2. If any Student is on leave on certain days and he enter his Id in mess, the server will realize he is on leave and show a message accordingly.

3. When the Mess Admin wants to view which mess workers are on leave on a certain day, the server will cross-reference the days of leave with current date and show the names accordingly.

2.3 User Characteristics

2.3.1 Authentication

The Student needs to enter his ID Number or scan his ID card to be authenticated. The Authentication server has an inbuilt functionality that alerts the mess admin in case the student logs in more than once for a particular meal, i.e, if the student attempts to eat the meal more than once, during the same meal session.

The Mess Crew and Mess Admin is expected can able to login using their unique username and password combinations.

The Guest can login directly by entering is name. His name will be stored in the database of people who have eaten the meal.

2.3.2 GUI

Our software will support a browser based GUI that is compatible with multiple operating systems supporting HTML. We will be using the Windows Builder plugin of the Eclipe to design this GUI.

2.4 Database

The data descriptions of each of these data entities are as follows:

2.4.1 Student Data Entity

Data Item	Type	Description	Comment
Sr No	Integer	Serial Number	
Name	String	Name	
ID	String	ID number	Used as username
Mess	String	Mess Option Selected	
HasEaten	Boolean	Whether the Student has eaten the current meal	Used during checkin
Password	String	Password for Login	
Leave Start	Date	Leave Start date	Only if applied for leave
Leave End	Date	Leave End date	Only if applied for leave
Dues	Array	Dues details	

2.4.2 Student Feedback Data Entity

Data Item	Type	Description	Comment
Sr No	Integer	Serial Number	
Name (Optional)	String	Name	
Comments	String	Feedback	

2.4.3 Guest Data Entry

Data Item	Type	Description	Comment
Sr No	Integer	Serial Number	
Name	String	Name of Guest	

2.4.4 Guest Feedback Data Entity

Data Item	Type	Description	Comment
Sr No	Integer	Serial Number	
Name (Optional)	String	Name	
Comments	String	Feedback	

2.4.5 (Mess Name)Mess Crew Data Entity for Each Corresponding Mess

Data Item	Type	Description	Comment
Sr No	Integer	Serial Number	
Name	String	Name of Mess Crew	
Username	String	Unique ID	Used to login
Password	String	Password for Login	
Mess	String	Mess Belonged to	
Job	String	Job Title	
Leave Start	Date	Leave Start date	Only if applied for leave
Leave End	Date	Leave End date	Only if applied for leave
Leave Status	Boolean	Whether Leave has been approved	Only if applied for leave
Account Balance	Double	Details of pay	
Shifts	String	Details of shifts	

2.4.6 (Mess Name)Mess Admin Data Entity for Each Corresponding Mess

Data Item	Type	Description	Comment
Sr No	Integer	Serial Number	
Name	String	Name of Mess Crew	
Username	String	Unique ID	Used to login
Password	String	Password for Login	
Mess	String	Mess Belonged to	
Leave Start	Date	Leave Start date	Only if applied for leave
Leave End	Date	Leave End date	Only if applied for leave
Account Balance	Double	Details of pay	
Shifts	String	Details of shifts	

2.4.7 (MessName)Mess Info Data Entity for Each Corresponding Mess

Data Item	Type	Description	Comment
Mess Name	String	Name of Mess	
Contractor Name	String	Name of Mess Contractor	
Breakfast Price	String	Cost of Breakfast	
Lunch Price	String	Cost of Lunch	
Dinner Price	String	Cost of Dinner	
Account Balance	String	Current Account Balance	

2.4.8 (MessName)Mess Menu Data Entity for Each Corresponding Mess

Data Item	Type	Description	Comment
Meal(Day)	String	Meals for Each Day	Eg. BreakfastMonday
Main Course Type	String	Veg / Non Veg	
Main Course	String	Main Meal	
Side	String	Accompaniments	
Salad	String	Accompaniments	
Beverage	String	Drinks	
Sweet	String	Sweet Dish	

2.4.9 (MessName)FoodStock

Data Item	Type	Description	Comment
Name	String	Name of Item	
Stock (in lots	Integer	Current stock of item	
Price	Integer	Price of FoodStock	

2.5 Security

The server on which the Mess Management System resides will have its own security to prevent unauthorized *read/write/edit* access.

The Admin’s account and the individual Mess Crew accounts are password protected. Only the Admin will have write/edit access to the server and the databases stored on it.

Student accounts need only a username for authentication to ensure smooth functioning of the mess.

3.0 Technology Implementation

3.1 Framework

We had the option to make our software desktop based or web based. We have chosen to develop it as an online platform to ensure compatibility with multiple systems. In doing so, we’ve effectively shifted the compatibility constraint from the OS to the browser.

We are using the following technologies to implement our software

1. Java JDK 7
2. HTML
3. Swing

We are using a plugin software called WindowBuilder Pro which is a powerful and easy to use bi-directional Java GUI designer that makes it very easy to create Java GUI applications without spending a lot of time writing code to display simple forms

3.2 Software Requirements

The following software is required for the development of the project:

1. Argouml (Version 0.34)
2. Eclipse (Luna)
3. WindowBuilder Pro for Eclipse (Version 1.7.4)
4. Microsoft Office (Version 2013)