My Desired Student Health Services System

Prepared for

Professor Alan R. Hevner

University of South Florida

Prepared by

Srirag Ramadugu

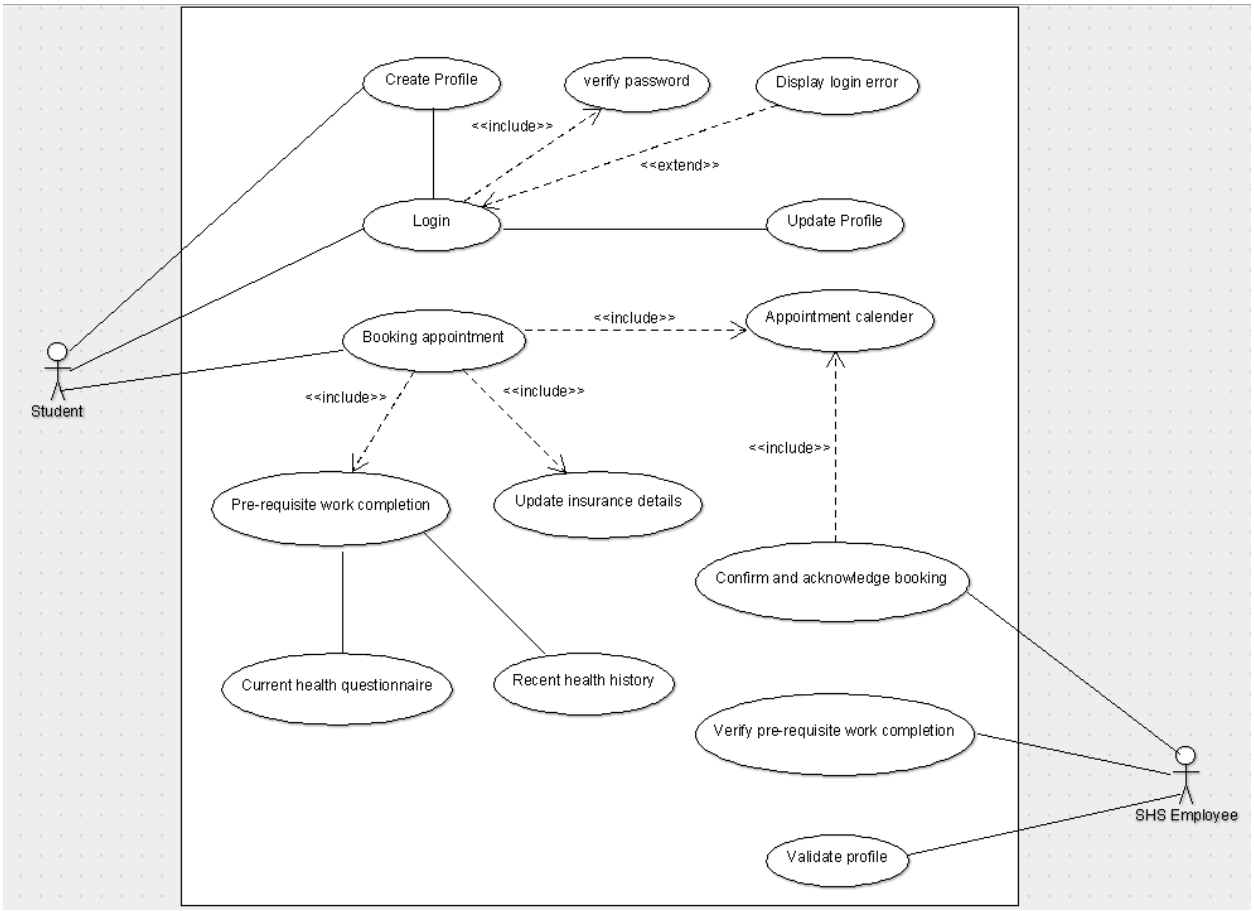
U23695579

October 11, 2018

Introduction:

This report is regarding assisting SHS (Student Health Services) in the requirements gathering and initial design of the software system that focuses on the basic student functions of entering and updating student profiles, booking appointments, and completing a current health questionnaire before each appointment. Use case diagram, Class diagram and State Chart diagram will be used to perform description of the desired system.

1. Use Case Diagram:



Use Case Diagram

Description:

Actors:

In any Use Case Diagram, an actor specifies a role played by a user or any other system that interacts with the subject. In the above diagram, student and SHS employee act as actors. Student is the primary actor for whom the software is being build. SHS Employee is the secondary actor who is going to work with the software to provide best services for the students.

Systems:

The rectangular box represents the system. In this context it represents the SHS system. Actors are placed outside the system and all the operations that are to be done in the software are included in systems.

Use Cases:

Use Cases represent the actions that are to be performed in collaboration with one or more external users of the system.

(Assumptions) Students Use Cases:

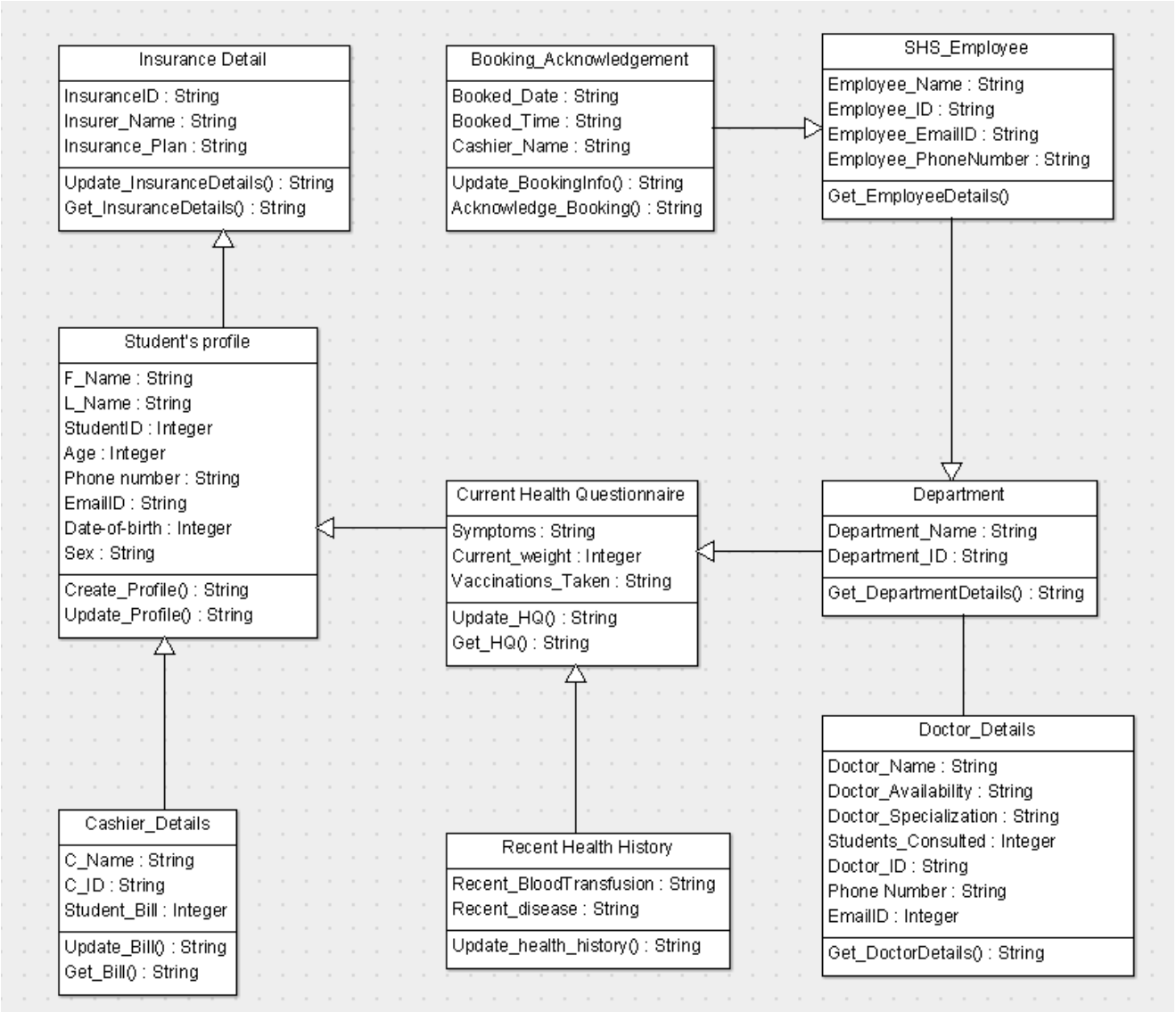
First student is supposed to create their profile using create profile. Then use Login use case to login to the page provided the password is verified. He can even update his profile once he gets logged-in. To book his appointment he should update his insurance details, pre-requisite work, recent health history and current health questionnaire. The system checks appointment calendar when the student tries to book an appointment.

(Assumptions) Employee Use Cases:

SHS Employee is supposed to verify students' profile, pre-requisite work completion and check calendar for an available date, and update and confirm appointment booking.

Include and extend relations are use in defining the relations as explained above. Include relationship is intended for reusing behavior modeled by another model. Extend relationship is intended for adding parts to existing use cases and modeling optional system services.

2. Class Diagram:



Outline of all the entities, their attributes, operations and associations:

Class Name	Attributes	Operations
Students Profile	1. F_Name:String 2. L_Name:String 3. StudentID:Integer 4. Age:Integer 5. Phone Number:String 6. EmailID:String 7. Date-of-birth:String 8. Sex:String	1. Update_Profile(Phone Number:String, EmailID:String) : String 2. Create_Profile()
Insurance Details	1. InsuranceID:String 2. Insurance_Name:String 3. Insurance_Plan:String	1. Update_InsuranceDetails(): String 2. Get_InsuranceDetails(): String
Current Health Questionnaire	1. Symptoms:String 2. Current_weight:String 3. Vaccinations_Taken:String	1. update_HQ():String 2. Get_HD():String
Recent Health History	1. Recent_BloodTransfusion : String 2. Recent_disease:String	1. Update_Health_History(): String
Department	1. Department_Name:String 2. Department_ID:String	1. Get_DepartmentDetails(): String
Doctor_Details	1. Doctor_Name:String 2. Doctor_Availability:String 3. Doctor_Specialization : String 4. Students_Consulted: Integer 5. Doctor_ID:String 6. Phone Number: String 7. EmailID:String	1. Get_DoctorDetails():String
SHS_Employee	1. Employee_Name:String 2. Employee_ID:String 3. Employee_EmailID: String 4. Employee_PhoneNumber : String	1. Get_EmployeeDetails: String
Booking Acknowledgement	1. Booked_Date:String 2. Booked_Time:String 3. Cashier_Name:String	1. Update_BookingInfo():String 2. Acknowledge_Booking() : String
Cashier_Details	1. Cashier_Details:String 2. C_ID:String 3. Student_Bill:String	1. Update_Bill():String 2. Get_Bill():String

Description:

Students' profile:

In students' profile class, all the students details like F_Name, L_Name, StudentID, Age, Phone Number, EmailID, Date-of-birth, Sex will be the attributes which will be updated by the student. For this to happen Create_Profile() operation is used.

Student can even update details like EmailID, phone number and age later. For this Update_Profile() operation is used. This is the parent class for Cashier_Details and Current Health Questionnaire class.

Insurance Details:

In Insurance Details class, students' insurance details like InsuranceID, Insurance_Name, Insurance_Plan will be the attributes that ill be updated by the students. This is possible using Update_InsuranceDetails() Operation and these details can be fetched by the employees later using the operation Get_InsuranceDetails().

As insurance details need the student profile information for the employee to know which insurance is used by a specific student, a generalization line(relation) is maintained.

Current Health Questionnaire:

Attributes like Symptoms, Current_weight and Vaccinations_Taken are used in Current Health Questionnaire class. These details are updated using update_HQ() operation and these details rea retrieved by the department for further analysis of the doctor, thus the generalization relation used.

Recent Health History:

Attributes used in this class are Recent_BloodTransfusion and Recent_disease which are updated by the students and is updated into the system using the operation Update_Health_History(). Current health questionnaire class data can be further used here to analyse students' illness and thus the generalization is used.

Department:

In Department class, Department_Name and Department_ID are used as attributes. Get_DepartmentDetails() operation is used to get this information for the employees after going through the Health Questionnaires.

Doctor_Details:

In Doctor_Details class, the details of the doctor like Doctor_Name, Doctor_Availability, Doctor_Specialization, Students_Consulted, Doctor_ID, Phone Number and EmailID are used as attributes. These details will be fetched by the department using the operation Get_DoctorDetails().

SHS_Employee:

The attributes Employee_Name, Employee_ID, Employee_EmailID, Employee_PhoneNumber are used in the class SHS_Employee. This class uses the data from Department class, about the details of the doctor available and once the health questionnaire is analyzed, suited department and available doctor is assigned and a booking is done.

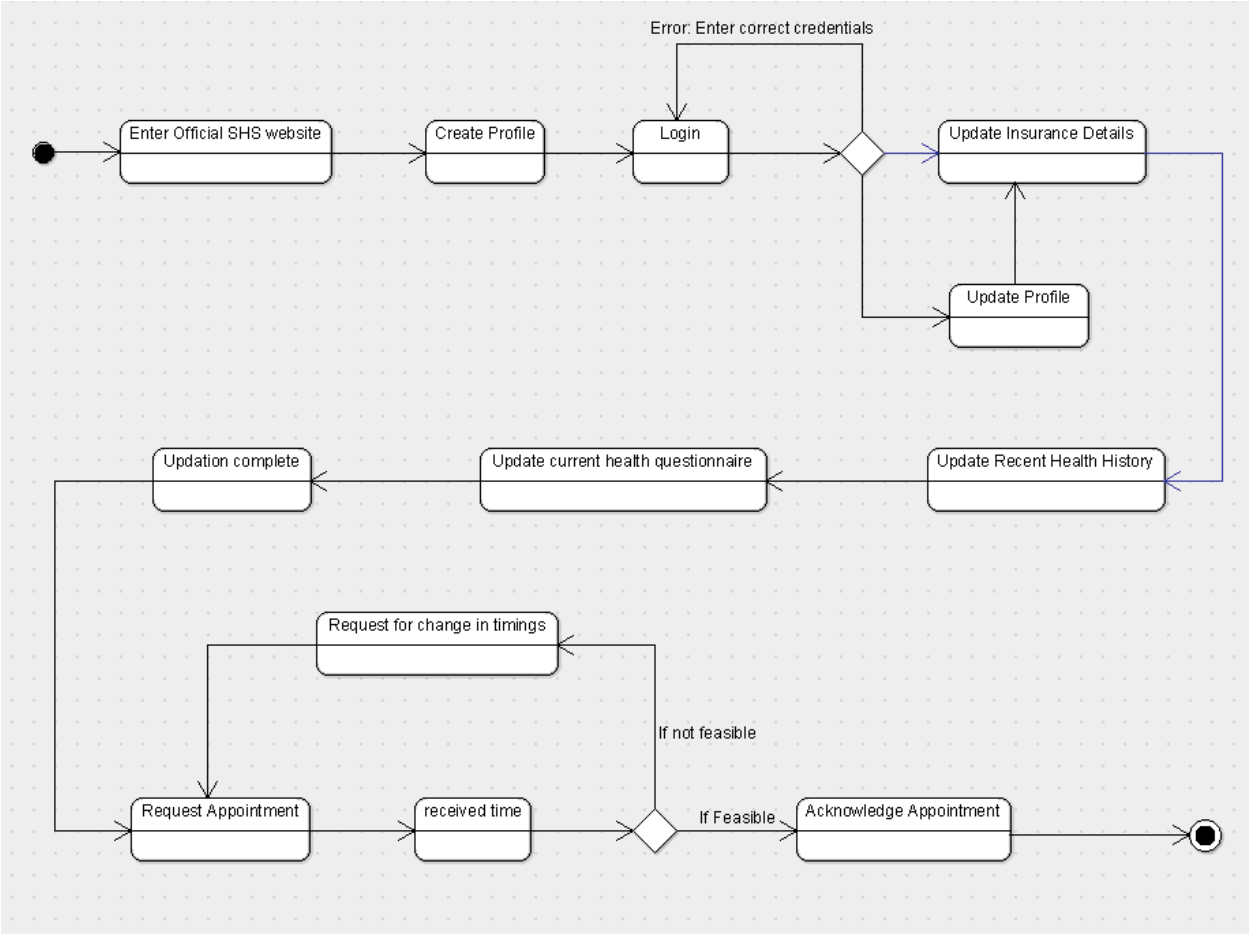
Booking Acknowledgement:

In Booking Acknowledgement class, the attributes used are Booked_Date, Booked_Time and Cashier_Name. Update_BookingInfo() operation is used by the employee to update the booking information onto the system. Acknowledge_Booking() operation is used to acknowledge the same to the student.

Cashier_Details:

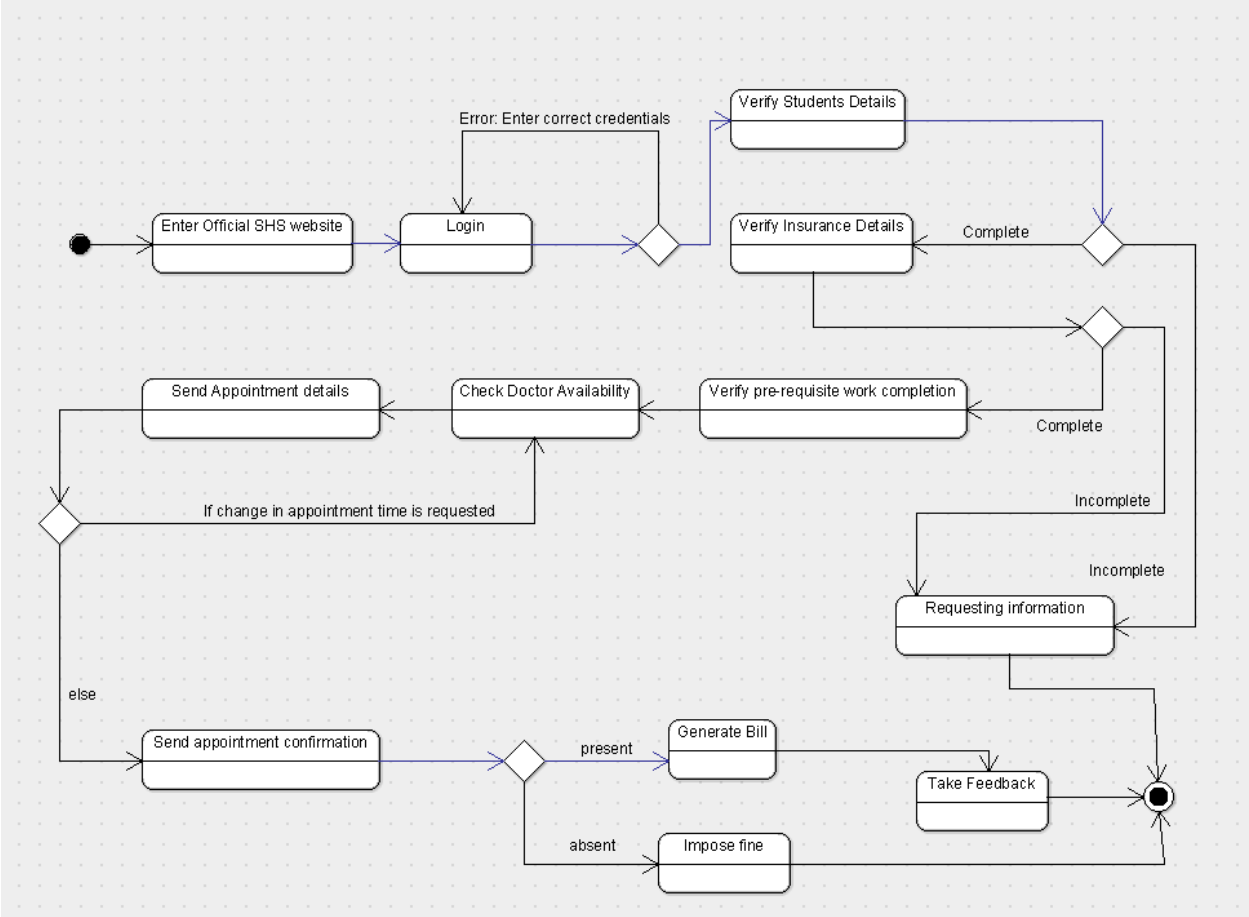
The attributes used in Cashier_Details class are C_Name, C_ID and Student_Bill. Update_Bill operation is used by the cashier to update the bill details into the system and Get_Bill operation is used by the cashier at the time of student check-out.

3. State Chart Diagram:



State chart diagram with respect to student

In the above state chart diagram with respect to student, it clearly shows my assumptions. Student first enters into official SHS website then create his profile, provided if it's his first time to use this service. He is supposed to login and we can see a condition where there is any login error then the flow goes back to login page displaying error message. If the student successfully logged-in then he should update his insurance details, if not updated earlier. He can even update his profile information like phone number and email-ID using update profile option. Then student need to update insurance details, recent health history details and current health questionnaire after which student need to request for appointment. Once the appointment time is received, student may check if that time is feasible or not and thereby acknowledge it.



State chart diagram with respect to SHS Employee

In the above state chart diagram with respect to SHS employee, it clearly shows my assumptions. First, they are supposed to enter the official website and then login with the password check like that of student. Then they should validate student details, insurance details and pre-requisite work completion. In this verification, if there is any incompletion, then a request for information is to be sent to the student. Based on the pre-requisite work, the department is chosen and thereby doctors availability is checked. Once the employee finds any availability, a mail is sent to the student regarding the same. If student requests for the change in appointment timings, then the employee is supposed to check for other timing, else, they are supposed to send appointment confirmation. If the student did show up for the appointment then after the consultation, generate the bill and take feedback from student. Else, impose a small amount of fine.