UNNC Clinic Management System (CMS)

Specification Sheet

1 Introduction

The UNNC-CMS system we created is a phone APP designed to offer services to university patients for booking their appointments before coming to the clinic. The target community includes students, employees, and their dependents and the expected number of users is around 30,000 persons. The doctors and nurses can also use it to view patients' appointments, record patients' data, write prescriptions, and make referrals to external hospitals. All users' information shall be stored in a separated database on the cloud for future use.

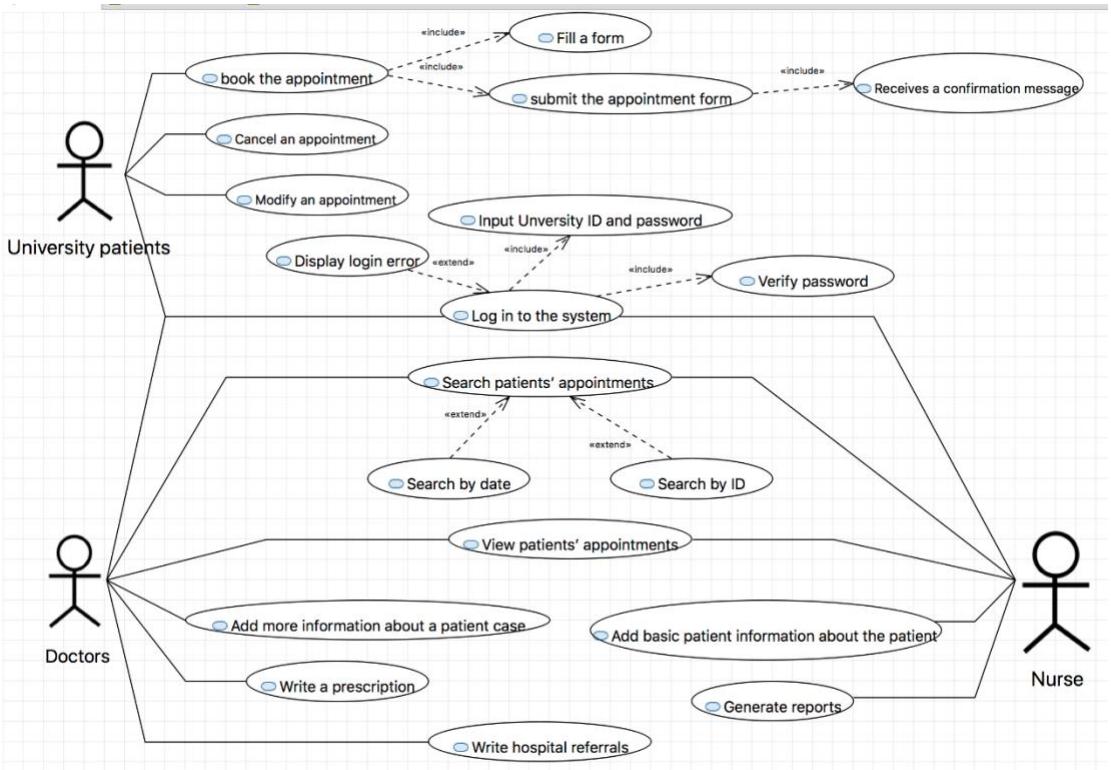
In our point of views, this project is better implemented using the incremental model due to the flexibility of this model. It is less costly to change scope and requirements. There are also some advantages to this model. Firstly, the incremental model is a model which can be tested and debugged easily during a smaller iteration. Thus, it is convenient to check the correctness between two phases in our project and manage risk. Another windfall of this model is that the risk can be easily managed due to the fact that risky pieces are identified and handled during its iteration and each iteration is an easily managed milestone. In that way, adopting this model can let customers respond to each phase and perform better with users' involvement. Lastly, lowering the initial delivery cost is also a reason for choosing this model.

There are five members in the developing team. A is our project manager, who is responsible for requirement analyze and project schedule management. UI design is responsible for B and C. As for IOS and Android development, D
and B will take responsibilities. Back-end Java server and database will be handled
by E. C is also accountable for Unit-test. Finally, daily maintenance will be carried out by team members in turn.

2 Personas, Actor and Action Table

Personas	**Actor**	**Usecase**
University patients	Student	Log in to the system
	Student	Fill a form
	Student	Submit a form
	Student	Receives a confirmation message
	Student	Modify an appointment (optional)
	Student	Cancel an appointment (optional)
	Employees	Log in to the system
	Employees	Fill a form
	Employees	Submit a form
	Employees	Receives a confirmation message
	Employees	Modify an appointment (optional)
	Employees	Cancel an appointment (optional)
	Dependents	Log in to the system
	Dependents	Fill a form
	Dependents	Submit a form
	Dependents	Receives a confirmation message
	Dependents	Modify an appointment (optional)
	Dependents	Cancel an appointment (optional)
Clinic	Nurses	Log in to the system
	Nurses	Search patients' appointments
	Nurses	View patients' appointments
	Nurses	Add basic patient information about the patient
	Nurses	Generate reports.
	Doctors	Log in to the system
	Doctors	Search patients' appointments
	Doctors	View patients' appointments
	Doctors	Add more information about a patient case,
	Doctors	Write a prescription
	Doctors	Write hospital referrals.

2 Usecase Diagram



3 User Requirements

This section contains all the users' functional requirements with regard to the emulator aspect of the system. Each requirement is prioritized as follows:

M Mandatory requirement. This feature must be built into the final system.

D Desirable requirement. This feature should be built into the final system unless its cost is too high.

O Optional requirement. This feature can be built into the final system at the Project Manager's discretion.

E Possible future enhancement. This feature is recorded here so that the idea is not lost. The decision on whether to include it in the system will depend on progress on the mandatory requirements.

Functional requirements

Label	Requirement	Necessity
1	The system needs to support multi phone platforms including IOS and Android.	**M**
2	The system can provide additional language support such as Chinese.	**D**
Login Requirement		
3.1	The system will display user terms of service when the first time of use.	**D**
3.2	The system shall allow users to log in with their university IT account.	**M**
3.3	The system can let users use fingerprint for further login after successful login via university IT account.	**O**
3.4	The system should provide "forget password" function to help users reset the password after identifying the user's identity when users forget the password.	**O**
Appointment Manage Requirement		
4.1	The system shall provide an appointment system, which including the creation of appointment, cancel and modify.	**M**
4.2	The system can let user append or export information from	**M**

	appointments.	
4.3	The system should show the number of bookings at a specific time period in order to cancel redundant bookings.	**D**
4.4	The system can alarm patients when the appointment is near.	**E**
4.5	The system can be linked to up-level hospitals for the direct hospital referrals and transfer information.	**D**
Appointment Retrieval Requirements		
5.1	The system should offer a function that shows appointments in a specific day.	**M**
5.2	The system needs to allow users with access search for a specific appointment according to student ID or name.	**M**
5.3	The system allows users to track history appointment and report of one patient.	**D**
User-Friendly Requirements		
6.1	The system allows patient pay fee online after diagnosing.	**E**
6.2	The system provides mechanisms for patients to rate service of the clinic and add comments.	**O**
6.3	The system should provide contact when users meets a problem with booking an appointment.	**D**

Non-functional requirements

Label	Requirement	Necessity
1	The system should ensure the privacy of user and security of data.	**M**
2	The system should be easy to learn and use.	**D**
3	The system should have a friendly and clear interface.	**D**
4	The system should backup data regularly in case of data loss.	**M**
5	No spam and other attracting advertisement in the app.	**M**

4 System Requirements

This section contains most of the system requirements. **Top level label number** is matched to user requirements.

The process environment for requirements are listed as follows:

C Client side, requirement(function) is processed locally and don't need to communicate with the server

S Server side, requirement(function) is processed by a remote server.

Label	Requirement	Process
Login Requirement		
3.1	The system shall display a form with **username** input, **password** input, **terms of use** checkbox, **language** dropdown list.	**C**
3.2	User can only click the "sign in" button when the form is filled correctly.	**C**
3.3	The system will check username and password in UNNC IT account database and return login status and user's group for showing the different user interface.	**S**
3.4	Extra language pack will be downloaded from the server if user chooses a language different from system language.	**S**
3.5	If login status 3.3 is false for 5 times, more attempts are prohibited. As a result, "sign in" button is changed to "forget password" button and linked to a password change website.	**C**
Appointment Management Requirement		
4.1	Normal User (patient mode)	
4.1.1	The system will query for unfinished appointment of current patient and return appointment data.	**S**
4.1.2	If unfinished appointment exists, then the system shall display the **basic information of current appointment(info)**, **"modify" button** and **"cancel" button**. If no unfinished appointment exists, then the system shall display **"new appointment" button**.	**C**

4.1.3	After clicking **"modify" button** or **" new appointment" button**, page for **form of appointment** will be displayed. For "modify" mode, there be current appointment information in the form, as for "new appointment mode", form is blank. And also, a **"confirm" button** will be displayed.	**C**
4.1.4	When user finishes editing and clicks the **"confirm" button**, modified or new appointment will be sent to server and server will send a confirm message to user's mailbox.	**S**
4.1.5	When user clicks the **"cancel" button**, an alarm window will pop and current appointment in database will be deleted if confirmed.	**C & S**
4.2	Super User (clinic mode)	
4.2.1	The system will display a textbox. The information can be appended in a textbox after the diagnose of the doctors. The nurse can add basic patient information about the patient such as attendance (present or absent), or patient's measurements (temperature, weight, and blood pressure) the measurements can have integer or real values while the doctor can add more information about a patient case and write a prescription.	**C**
4.2.2	The system will display a save button after the textbox. By clicking it, the information of the patients and the prescription can be saved. A clicking a button named 'Export the report' is also displayed. The user can click it to print out a report.	**C & S**
4.2.3	The system can be linked to up-level hospitals for direct hospital referrals and transfer information. Specifically, the system should provide a "submit to up-level database" button which is used to transfer the diagnosis's information to up-level hospitals' database after doctors or nurses finish the referrals.	**C & S**
Appointment Retrieval Requirement		
5.1.1	When nurse logins to the system, search box is displayed, and user	**C**

	is asked to enter a 6-digit date.	
5.1.2	Once date is selected, the system will return all appointment within chosen data and displayed as list box.	**C & S**
5.2	When nurse or doctor logins to the system, patient ID or name can be entered and the current appointment for this patient will be searched and displayed. There will also another **"history for this patient" button** being displayed.	**C & S**
5.3	When user clicks **"history for this patient" button**, history appointment will be requested from server and displayed as list box.	**C & S**
User-Friendly Requirements		
6.1.1	If there is previous finished appointment, a rating bar and a text box for comment will be displayed.	**C**
6.1.2	After being rated and commented by patient, the evaluation will be updated to database for record.	**S**
6.2	All communication between client and server should be encrypted.	**C & S**
6.3	Server side will backup all database and upload the backup file to another server every day 24:00 in case of data loss	**S**

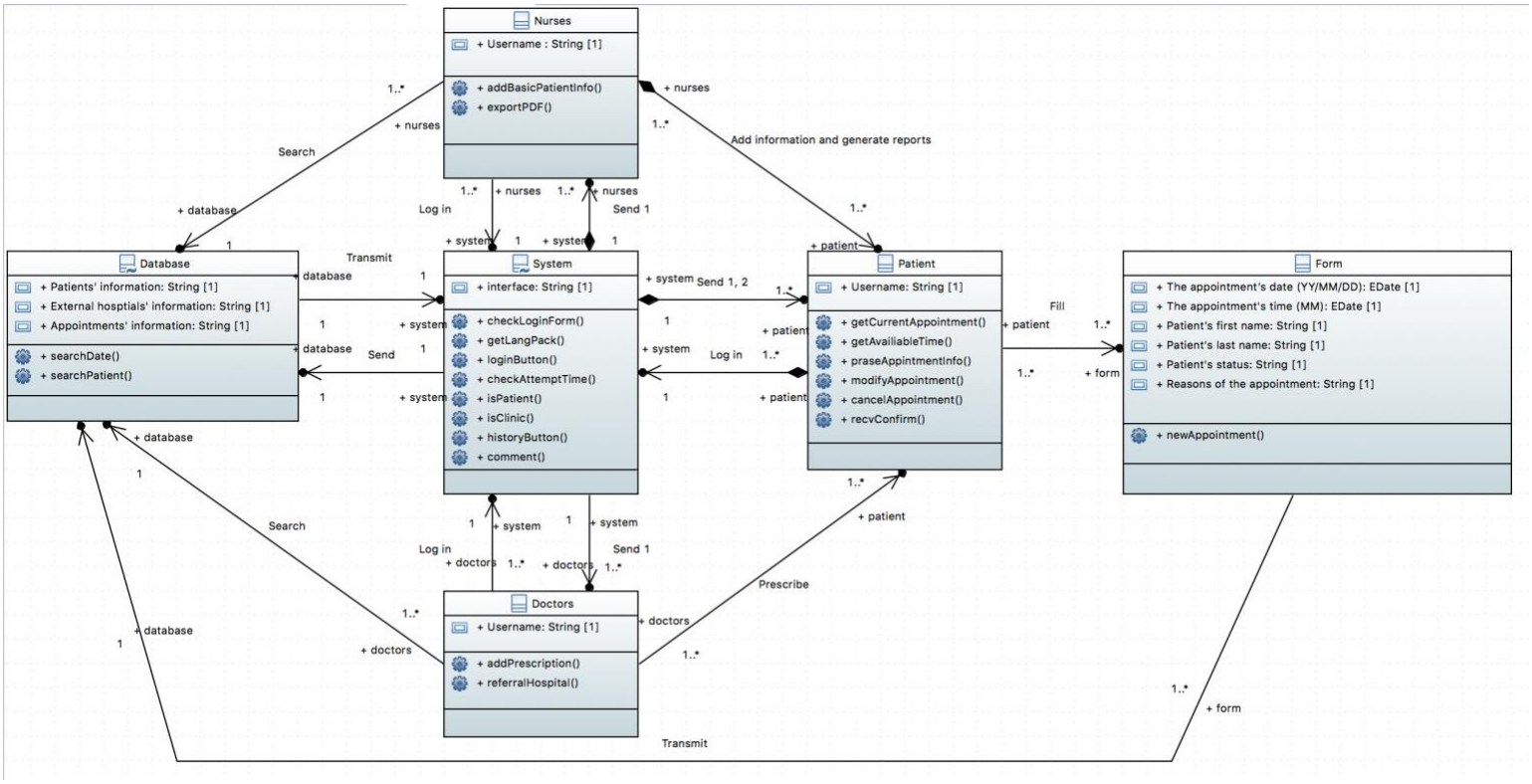
5 Method List

Method name	Input	Type	Output	Type
Login Requirement				
checkLoginForm()	username	String	loginButtonFlag	Boolean
	password	String		
	language	String		
	termFlag	Boolean		
getLangPack()	language	String	languagePack	String
loginButton()	username	String	loginStatus	Boolean
	password	String	userGroup	String
checkAttemptTime()	num_try	Int	loginButtonFlag	Boolean
			forgetButtonFlag	Boolean
Appointment Management Requirement				
isPatient()	userGroup	String	patientFlag	Boolean
isClinic()	userGroup	String	clinicFlag	Boolean
Normal User (patient mode)				
getCurrentAppointment()	username	String	appointmentInfo	String
getAvailiableTime()			availiableTime	String
newAppointment()	patientFirstName	String	createFlag	Boolean
	patientLastName	String		
	chosenTime	String		
	patientStatus	String		
	appointmentReason	String		
recvConfirm()	createFlag	Boolean	confirmMsg	String
praseAppintmentInfo()	appointmentInfo	String	appointmentExist	Boolean
			appointmentID	String
			patientFirstName	String
			patientLastName	String

			chosenTime	String
			status	String
			reason	String
modifyAppointment()	appointmentID	String	updateFlag	Boolean
	patientFirstName	String		
	patientLastName	String		
	chosenTime	String		
	patientStatus	String		
	appointmentReason	String		
cancelAppointment()	appointmentID	String	appointmentFlag	Boolean
Super User (clinic mode)				
addBasicPatientInfo()	appointmentID	String	updateFlag	Boolean
	appointmentAttend	Boolean		
	patientMeasurements	Double		
addPrescription()	appointmentID	String	updateFlag	Boolean
	patientPrescription	String		
exportPDF()	appointmentID	String	fileLink	String
referralHospital()	appointmentID	String	referralFlag	Boolean
	hospitalID	String		
Appointment Retrieval Requirement				
searchDate()	appointmentDate	String	appointmentID	String
			patientFirstName	String
			patientLastName	String
			chosenTime	String
			status	String
			reason	String
searchPatient()	username	String	appointmentID	String
	patientFirstName	String	patientFirstName	String

	patientLastName	String	patientLastName	String
			chosenTime	String
			status	String
			reason	String
historyButton()	historyFlag	Boolean	appointmentID	String
			patientFirstName	String
			patientLastName	String
			chosenTime	String
			status	String
			reason	String
User-Friendly Requirements				
comment()	rating	Int	commentFlag	Boolean
	comment	String		

6 Class Diagram



We omit some details in this diagram due to the limitation of the space. The details will be explained as follows:

Patient	three categories including students, employees and dependents.
Send 1	error message when logging into the system
Send 2	confirm message after filling the forms
Send & Transmit	the interaction between two class diagrams
Search	the search function is executed in the database class and the result can be viewed by clinic operators