SOFTWARE REQUIREMENT SPECIFICATION

[MEDIPOINT]

VERSION: [2.0]   REVISION DATE: [April 18, 2015]

Approver Name	Title	Signature	Date

SCHOOL OF COMPUTER ENGINEERING

NANYANG TECHNOLOGICAL UNIVERSITY

Table of Contents

1. Product Description

1.1. Purpose

A large private medical practice has a number of specialist clinincs in Singapore, Malaysia and Thailand and it offers consultations and outpatient medical services, including day surgery. The medical practice wants to expand its network and offer a wide rane of medical services. Their senior management wishes to launch an App for mobile devices to allow patients to manage their appointments with consultants and their related treatments.

The main purpose of this project is to build an android application which fulfils the requirements set by the management of medical practice and provides a convenient way to patients for booking and managing their appointments and related treatments with the consultants.

1.2. Scope

The software application will provide a platform to patients to manage their appointments on the go without having the trouble of contacting the hospital staff or physically going there. The application will facilitate an easy to use interface to patients, for them to keep track of their appointments, personal information and medical history. The application database will be maintained on a centraised searver maintained by the IT admin of the medical practice or their contractors to eBnsure real time updates about consultant availability to the patients.

1.3. Stakeholders and Users

The **stakeholders** for this project will be the developers of the application who put in efforts and resources to develop a suitable application for the medical practice.

The **users** of this application will be the **medical practice staff** and their **patients**. Patients will be able to manage their appointments using the application while the hospital staf will be able to view the appointments of patients and their confidential medical history through secure corporate accont s. In nutshell, the application will be an intermediate between the patients and the hospital staff and it will eliminate the need of maintaining physical records for managing appointments and information about related treatments.

1.4. Assumptions

1.4.1. No fake clients.

1.4.2. A doctor works everyday in one clinic from 9 am to 9 pm.

1.4.3. The user of the application are connected to Internet while using the application to update information in centtralised database server.

1.4.4. The general pre-appointment actions are based on the service type and any specific tests or actions that need to be done before the appointment are conveyed to the patient through email or SMS reminders.

1.4.5. There is a limit on number of cancellations a user can make per day.

1.5. Constraints

Limited amount of time and experience in app development will lead to limited functionalities and testing.

2. *Functional Requirements*

2.1. Managing User Accounts

2.1.1. Sign Up

2.1.1.1. The system displays sign up form.

2.1.1.2. The user must enter his Name.

2.1.1.3. The user must enter his NRIC Number.

2.1.1.3.1. The system displays an error message if NRIC is invalid or already exists.

2.1.1.4. The user must enter his E-mail.

2.1.1.4.1. The system displays an error message if email is invalid.

2.1.1.5. The user must enter his Contact Number.

2.1.1.5.1. The system generates an error message if Contact Number is invalid.

2.1.1.6. The user must enter his Address.

2.1.1.7. The user must select his Citizenship.

2.1.1.8. The user must select his Country of Residence.

2.1.1.9. The user must select his/her Gender.

2.1.1.10. The user must select his Marital Status.

2.1.1.11. The user must select his Preferred Notification Mode.

2.1.1.12. The user must select his Date of Birth.

2.1.1.12.1. The system displays an error message if Date of Birth is after the current date.

2.1.1.13. The user must enter a Username.

2.1.1.13.1. The system displays an error message if username is already taken.

2.1.1.14. The user must enter a Password.

2.1.1.15. The user must confirm the entered password.

2.1.1.15.1. The system displays an error message if Confirm Password does not match with entered Password.

2.1.1.16. The system creates an account.

2.1.1.17. The system displays success message.

2.1.1.18. The system displays Medical History form.

2.1.1.19. The user fills in the Personal Hisotry, Drug Allergy, Ongoing Treatment and Ongoing Medication.

2.1.1.20. The system generates success response.

2.1.1.21. The system redirects user to login interface.

2.1.2. Sign In

2.1.2.1. The user enters his Username.

2.1.2.2. The user enters his account Password.

2.1.2.3. The system verifies the Username and Password.

2.1.2.3.1. The system generates an error message if Username or Password is invalid.

2.1.2.4. The system generates welcome message and redirects the user to Homepage.

2.1.3. Sign Out

2.1.3.1. The user selects Logout option.

2.1.3.2. The system logs out the user and redirects the user to login page.

2.1.4. Forgot Password

2.1.4.1. The user selects Forgot Password option.

2.1.4.2. The system redirects user to Password Retreiver interface.

2.1.4.3. The user enters his NRIC and confirms.

2.1.4.4. The system verifies the NRIC.

2.1.4.4.1. The system generates an error message if NRIC does not exists.

2.1.4.5. The system sends the password to user’s email.

2.1.4.6. The system redirects user to login page.

2.2. Manage Medical Details of the User

2.2.1. The user selects Medical History option.

2.2.2. The system displays Personal Information of the user.

2.2.3. The system displays Medical History of the user.

2.2.4. The system displays Drug Allergy information of the user.

2.2.5. The system displays information regarding Ongoing Treatment and Medications of the user.

2.2.6. The user chooses to edit the medical details.

2.2.6.1. The system retrieves the medical details of the user.

2.2.6.2. The system reloads the form with retrieved details already selected.

2.2.6.3. The user can add or remove details.

2.2.6.4. The system updates the medical details.

2.2.6.5. The system generates a confirmation message and redirects user to Homepage.

2.3. Send Reminders

2.3.1. The system sends reminder via preferred mode of notification 24 hours before the upcoming appointment.

2.3.1.1. The reminder must contain date and time of appointment.

2.3.1.2. The reminder must contain the specialization and service type.

2.3.1.3. The reminder must contain the name of the doctor and any suggestions made by the doctor.

2.3.1.4. The reminder must contain information about any pre-appointment actions or suggestions from doctor.

2.4. Manage an Appointment

2.4.1. Create appointment

2.4.1.1. The user must be able to book an appointment for atleast a year in advance.

2.4.1.2. The user must select if it's a new appointment, referral or a follow up.

2.4.1.3. The user selects a **New Appointment**.

2.4.1.3.1. The user must select a Country.

2.4.1.3.2. The user must select a Clinic.

2.4.1.3.2.1. The Clinic location depends on the the Country selected.

2.4.1.3.3. The user must select a Specialty.

2.4.1.3.3.1. The Specialty options can depend on the clinic location.

2.4.1.3.4. The user must select a Service Type.

2.4.1.3.4.1. The service type depends on the Specialty selected.

2.4.1.3.5. The user can select a Doctor Name.

2.4.1.3.5.1. The doctor names depend on the clinic location and specialty.

2.4.1.3.6. The user must select a date for the appointment.

2.4.1.3.6.1. The appointment date must be atleast 24 hours from the current date.

2.4.1.3.7. The user must select a time slot.

2.4.1.3.7.1. The system must only show slots which are available depending on doctor availability and user's other appointments.

2.4.1.3.7.2. The available slots must be shown in list form by the system.

2.4.1.3.7.3. The slot must be atleast 30 minutes long.

2.4.1.3.7.4. The slot duration depends on the Service selected.

2.4.1.3.7.5. The system must reset the time slot if any other option is changed.

2.4.1.3.8. The system books an appointment for the user.

2.4.1.3.9. The system generates confirmation message and returns user to Homepage.

2.4.1.3.10. The list of appointments is displayed with key information.

2.4.1.4. Appointment is a **Referral**.

2.4.1.4.1. The user must select a Country.

2.4.1.4.2. The user must select a Clinic which has referred the user.

2.4.1.4.3. The user must select a specialty and doctor who gave the referral.

2.4.1.4.4. The user follows the steps under 2.4.1.3.

2.4.1.5. Appointment is **Follow-Up**.

2.4.1.5.1. The user must have a past appointment.

2.4.1.5.2. The user must select a finished appointment to follow up.

2.4.1.5.3. The system must retrieve the required information for the selected appointment.

2.4.1.5.3.1. The system retrieves country where appointment was held.

2.4.1.5.3.2. The system retrieves clinic location.

2.4.1.5.3.3. The system retrieves specialty.

2.4.1.5.3.4. The system retrieves doctor name.

2.4.1.5.4. The user must select a Service type for follow up.

2.4.1.5.5. The user must select a date for the appointment.

2.4.1.5.5.1. The appointment date must atleast be 24 hours from the current date.

2.4.1.5.6. The user must select a time slot.

2.4.1.5.6.1. The system must only show the available slots.

2.4.1.5.7. The system books an appointment for the user.

2.4.1.5.8. The system generates a confirmation and redirect user to the appointment list view page.

2.5. View Appointment

2.5.1. The system must display a list of finished and upcoming appointments on the Homepage.

2.5.2. The user can select an appointment to view details.

2.5.3. The system retrieves appointment information.

2.5.4. The system redirects user to the View appointment interface.

2.5.5. The detailed appointment information is displayed.

2.5.6. Change/Cancel an Appointment

2.5.6.1. The user must select an appointment to edit from the appointment list on homepage.

2.5.6.2. The system provide an option to edit or delete an appointment.

2.5.6.3. The user **changes the appointment**.

2.5.6.3.1. The system retrieves the appointment information.

2.5.6.3.2. The user can change the Country.

2.5.6.3.3. The user can change the Clinic Location.

2.5.6.3.4. The user can change the Specialty.

2.5.6.3.5. The user can change the Service type.

2.5.6.3.6. The user can change the doctor to meet for the appointment.

2.5.6.3.7. The user can change the date for the appointment.

2.5.6.3.8. The user can change the time slot for the appointment.

2.5.6.3.9. The user must confirm the changes.

2.5.6.3.10. The system updates the appointment details.

2.5.6.3.11. The system generates a confirmation message and redirects the user to appointment list page.

2.5.6.4. The user **cancels the appointment**.

2.5.6.4.1. The system displays the appointment information.

2.5.6.4.2. The system asks for confirmation.

2.5.6.4.3. The user confirms.

2.5.6.4.3.1. The system cancels the appointment.

2.5.6.4.3.2. The system generates a confirmation message and redirects the user to appointment list page.

2.5.6.4.4. The user did not confirm.

2.5.6.4.4.1. The system return to appointment information display page.

3. Non-Functional Requirements

3.1. Usability

3.1.1. The system must be easy to learn and use.

3.1.1.1. The user must be able to use the system with minimal instructions.

3.1.1.2. The user must be able to retain the usage mechanism.

3.1.2. The system must respond to user action within few seconds.

3.1.3. The system must have consistent design.

3.1.3.1. The buttons must be well sized to comfort user's visual capabilities.

3.1.3.2. The text must be well sized to comfort user's visual capabilities.

3.1.3.3. The design must be clear and visually appeal to the user.

3.1.4. The system must offer informative feedback for every user action.

3.1.4.1. The system must generate error messages for invalid inputs.

3.1.4.2. The system must organize action sequqneces into organized groups with a beginning, middle and end.

3.1.4.2.1. The system must generate confirmation messages for each of the stages.

3.1.4.3. The system must seek confirmation from the user for actions such as deletion or editing.

3.1.5. The system must offer simple error handling.

3.1.5.1. The system design must implement dropdowns, radio or checkbox selections to minimize chances of error.

3.1.6. The system must permit easy reversal of actions for users to know that errors can be undone.

3.2. Reliability

3.2.1. The system must be able to store information correctly into the database.

3.2.1.1. The system must be able to perform CRUD operations on the databse accurately.

3.2.2. The system must achieved 99.99% uptime.

3.2.3. The system must retain the user data for atleast 10 years.

3.2.4. The system must be secure and ensure user privacy.

3.2.4.1. The system shall not reveal any sensitive information to any third party unless stated.

3.2.5. After maintenance the system must achieve full functionality within 30 seconds.

3.3. Performance

3.3.1. On user login, the system must finish authenticiation within 30 seconds.

3.3.2. The application must not have response time exceeding 2 minutes by utilizing adequate device reources as needed.

3.3.3. The system must update database within 30 seconds.

3.3.4. After a user logs in, the system must finish the authentication at most 45 seconds.

3.3.5. The system must send notification to the user regarding the appointment.

3.3.5.1. System must sends a confirmation message within 30 seconds after the appointment has been placed.

3.3.5.2. System must sends a reminder within 1 day before the appointment.

3.4. Supportability

3.4.1. The system must support a wide variety of database systems.

3.4.2. The system must be able to perform all functionalities on different Android devices.

3.4.2.1. The system must be able to perform all functionalities in Android operating system.

4. Interface Requirements

4.1. User Interface

4.1.1. The interface must be easy to use with minimal instructions.

4.1.2. The user must be in control.

4.1.3. The navigation bar must be clear and consistent.

4.1.4. The size of text labels, buttons and other widgets must comfort user's visual capabilities.

4.1.5. The design must allow easy reversal of actions.

4.1.6. The design must be consistent and follow Android Design Guidelines.

4.1.7. The menu buttons, slide bars, dropdown lists and pop up windows must adhere to Android Standard.

4.1.8. The Lo-Fi sketch for the user interface is shown in figure 1.

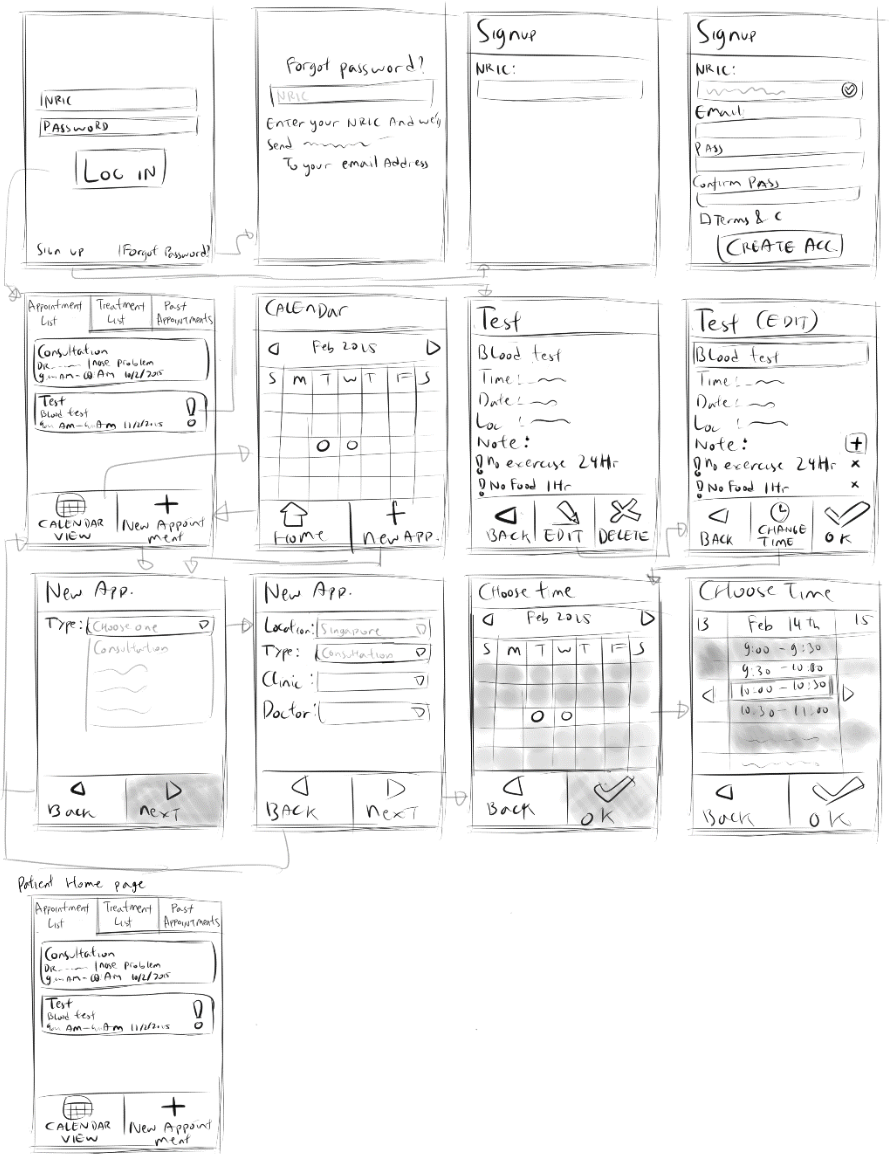

Figure 1

4.2. Hardware Interfaces

4.2.1. The application is hosted on an Android device. Its does not require any interface with any other external hardware.

4.3. Software Interfaces

4.3.1. The system must be able to interact with Android Notification system to set and trigger the notification ate the accurate time.

4.3.2. The system must be able to interact with Java Mail library to send reminders to users via email or SMS.

5. Use Case Model

Provide the top-level use case diagram, followed by the use case description for each use case.

5.1. Use Case Diagram – MediPoint Application

The figure 2 shows the overall use case diagram for MediPoint. The part colored in green indicated the functionalities which will be implemented in future and have not been included the current application.

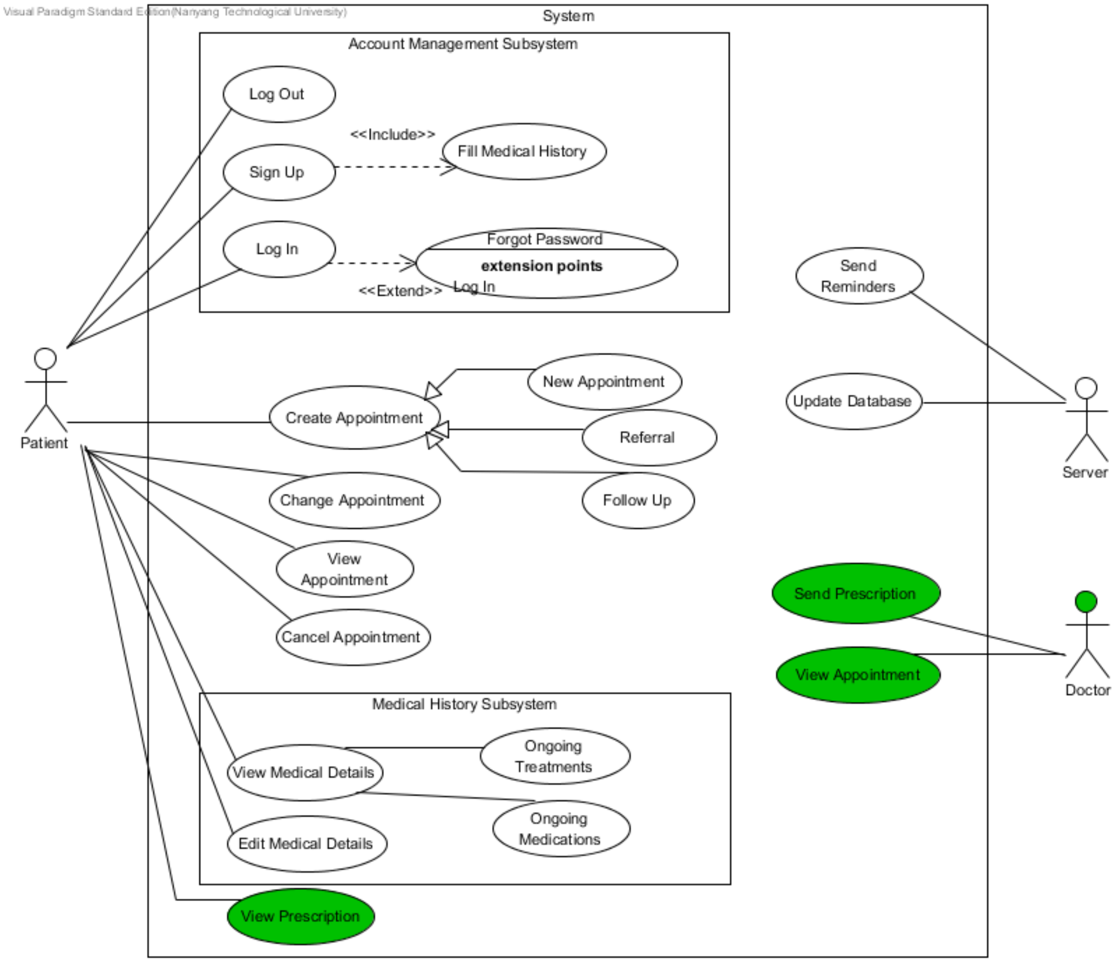

Figure 2

5.2. Account Management Subsystem

5.2.1. Use Case Model

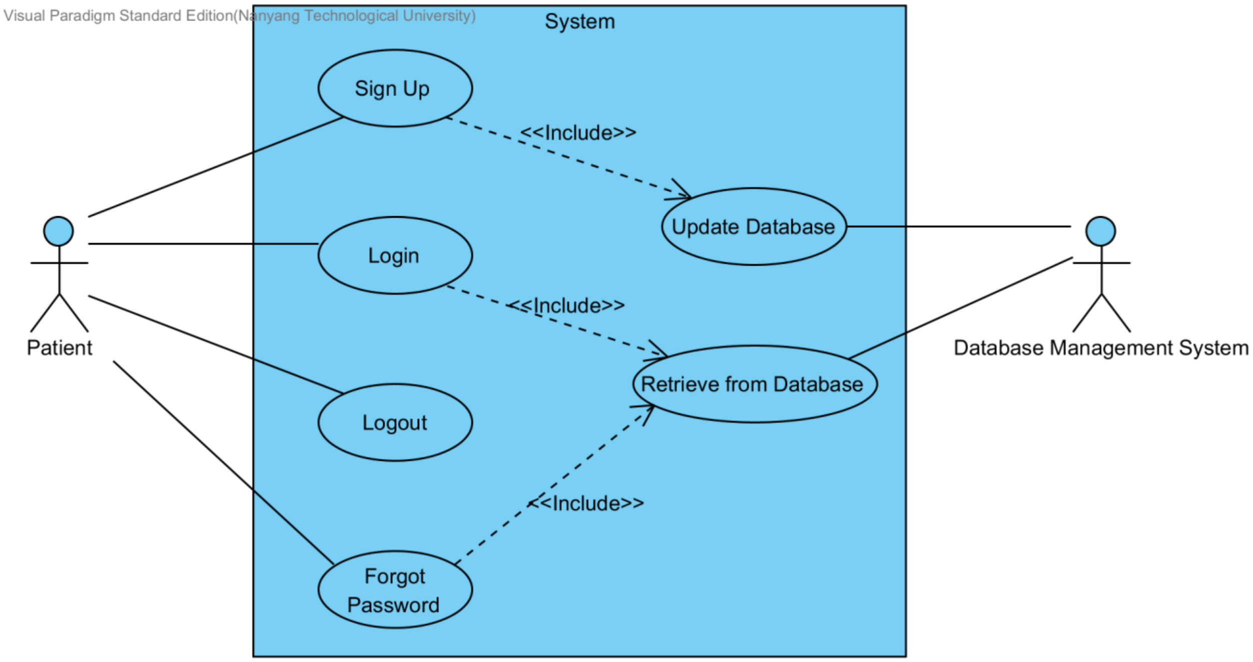

Figure 3

5.2.2. Use Case Description - Sign Up

Use Case ID:	1.1		
Use Case Name:	Sign up		
Created By:	Ankur	Last Updated By:	Shreyas
Date Created:	February 6, 2015	Date Last Updated:	April 17, 2015
Actor:	Patient, Database Management System		
Description:	Patient inputs personal information to create an account.		
Preconditions:	-		
Postconditions:	Patient is directed to fill medical history form.		
Priority:	2		
Frequency of Use:	Once per user (Recommended).		
Flow of Events:	1. Patient enters his full name, NRIC, E-mail, Contact No., Address, Citizenship, Country of Residence, Gender, Marital Status, Preferred Notification Mode, DOB, Username and Password. 2. Patient submits the information. 3. Update Database. 4. System confirms account creation.		
Alternative Flows:			
Exceptions:	1.1.EX.1 NRIC already exists : Patient gets an alert that account already exists and system		

	exits to Login. 1.1.EX.2 NRIC is invalid: System requests patient to enter a valid NRIC. 1.1.EX.3 Email address is invalid: System requests patient to enter a valid email address. 1.1.EX.4 Contact number is invalid: System requests patient to enter a valid contact number. 1.1.EX.5 Incomplete details: System requests patient to fill all the fields. 1.1.EX.6 Incorrect date of birth: System requests the patient to enter his date of birth correctly. 1.1.EX.7 Username already exists: System notifies the patient that the username already exists. 1.1.EX.8 Confirmed password is incorrect: System notifies the patient that the confirmed password is incorrect.
Includes:	Update Database.
Special Requirements:	1. User must enter all the details before proceeding further at any stage of account creation.
Assumptions:	1. All the details entered by the user are accurate.
Notes and Issues:	

5.2.3. Use Case Description - Sign In

Use Case ID:	1.2		
Use Case Name:	Sign in		
Created By:	Ankur	Last Updated By:	Shreyas
Date Created:	February 6, 2015	Date Last Updated:	April 17, 2015
Actor:	Patient, Database Management System.		
Description:	Patient inputs username and password to login.		
Preconditions:			
Postconditions:			
Priority:	2		
Frequency of Use:	Every time user uses the application.		
Flow of Events:	1. Patient enters his Username and Password. 2. Retrieve From Database. 3. System creates a session. 4. Patient is redirected to his homepage.		
Alternative Flows:			
Exceptions:	1.2.EX.1 If Username does not exist in the database or password does not match, an error message is displayed.		
Includes:	Retrieve From Database		
Special Requirements:			
Assumptions:			
Notes and Issues:			

5.2.4. Use Case Description – Sign Out

Use Case ID:	1.3		
Use Case Name:	Sign out		
Created By:	Ankur	Last Updated By:	Shreyas
Date Created:	February 6, 2015	Date Last Updated:	April 17, 2015
Actor:	Patient		
Description:			
Preconditions:	Patient must be signed in		
Postconditions:			
Priority:	2		
Frequency of Use:			
Flow of Events:	1. System deletes the patient’s session. 2. Patient is redirected to the login page.		
Alternative Flows:			
Exceptions:			
Includes:			
Special Requirements:			
Assumptions:			
Notes and Issues:			

5.2.5. Use Case Description – Forgot Password

Use Case ID:	1.4		
Use Case Name:	Forgot Password		
Created By:	Shreyas	Last Updated By:	Shreyas
Date Created:	February 6, 2015	Date Last Updated:	April 18, 2015
Actor:	Patient, Database Management System, Notification Management System		
Description:	Patient is sent his forgotten password by email		
Preconditions:	1. Patient should already have an existing account		
Postconditions:	1. Patient has received his password by email		
Priority:	3		
Frequency of Use:			
Flow of Events:	1. Patient enters his NRIC 2. Retrieve From Database (Patient’s email and password are retrieved in this step). 3. Notification Management System sends the patient his password by email. 4. Patient is redirected to the login page.		
Alternative Flows:			
Exceptions:	1.4.EX.1 NRIC not found: The patient is notified that he does not have any existing account.		
Includes:	Retrieve From Database		
Special Requirements:			
Assumptions:	Patient enters his correct NRIC		
Notes and Issues:			

5.3. Appointment Management Subsystem

5.3.1. Use Case Model

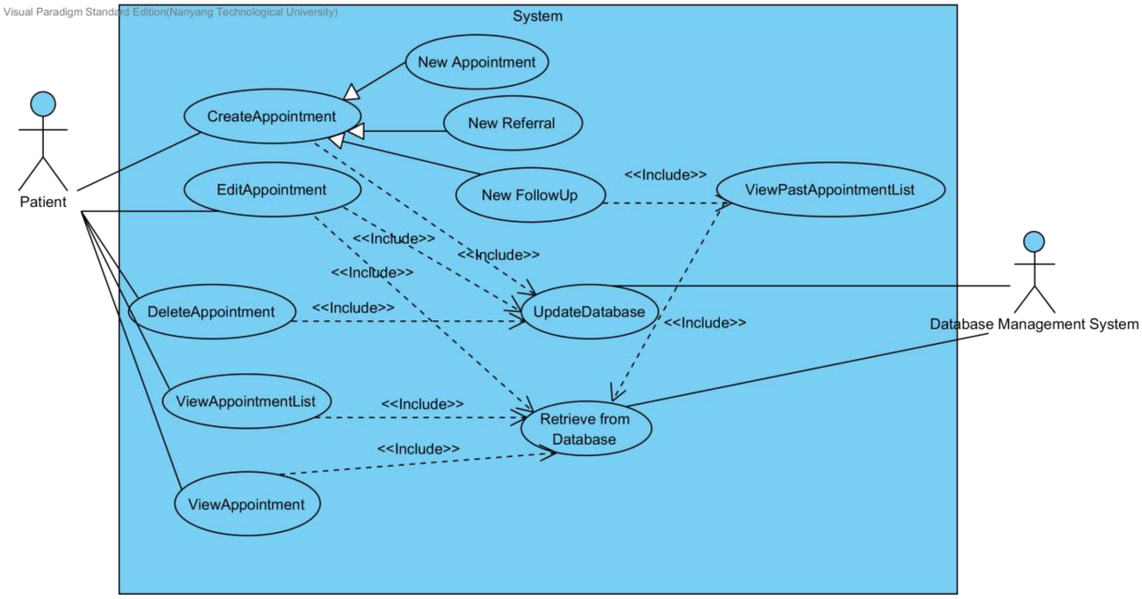

Figure 4

5.3.2. Use Case Description – Create Appointment

Use Case ID:	2.1		
Use Case Name:	Create Appointment		
Created By:	Deka	Last Updated By:	Ankur
Date Created:	February 6, 2015	Date Last Updated:	April 18, 2015
Actor:	Patient, Database Management System, Notification Management System		
Description:	Creating new appointment		
Preconditions:	Patient has an account and has logged in to the application		
Postconditions:	Patient has been notified about his upcoming appointment.		
Priority:	1		
Frequency of Use:	Twice a month		
Flow of Events:	1. Patient selects his country, clinic location, specialty, service, doctor, appointment date and selects the available appointment time and confirms the appointment. 2. Update Database. 3. Notification Management System sends a notification to the user about the upcoming appointment. 4. Notification Management System sends an email and/or SMS to the user about the upcoming appointment depending on his preferred mode of notification. 5. The patient is redirected back to the home page.		
Alternative Flows:			
Exceptions:	2.1.EX.1 User does not select an appointment date: System requests the user to select a date.		

	2.1.EX.2 User does not select the appointment time: System requests the user to select a time. 2.1.EX.3 User selects a past date or a date less than 24 hours after the current date: System requests the user to book at least 24 hours in advance.
Includes:	Update Database
Special Requirements:	
Assumptions:	1. The patient had entered his correct email address and contact number during registration. 2. The patient will complete any pre-appointment tests or actions before the appointment.
Notes and Issues:	

5.3.3. Use Case Description – Create Referral

Use Case ID:	2.2		
Use Case Name:	Create Referral Appointment		
Created By:	Deka	Last Updated By:	Ankur
Date Created:	February 10, 2015	Date Last Updated:	April 17, 2015
Actor:	Patient, Database Management System, Notification Management System		
Description:	Creating referral appointment		
Preconditions:	Patient has an account and has logged in to the application		
Postconditions:	Patient has been notified about his upcoming appointment.		
Priority:	2		
Frequency of Use:	Twice a month		
Flow of Events:	1. Patient selects the country, clinic location, specialty, and doctor which referred him. 2. Create Appointment.		
Alternative Flows:			
Exceptions:			
Includes:	Create Appointment		
Special Requirements:			
Assumptions:	1. The patient will be responsible for bringing the referral letter to the clinic on the day of appointment.		
Notes and Issues:			

5.3.4. Use Case Description – Create Follow Up

Use Case ID:	2.3		
Use Case Name:	Create Follow Up Appointment		
Created By:	Deka	Last Updated By:	Ankur
Date Created:	February 6, 2015	Date Last Updated:	April 18, 2015
Actor:	Patient, Database Management System, Notification Management System		
Description:	Creating follow up appointment		

Preconditions:	Patient has an account and has logged in to the application
Postconditions:	Patient has been notified about his upcoming appointment and is redirected to the homepage upon successful follow up appointment creation
Priority:	2
Frequency of Use:	Twice a month
Flow of Events:	1. View Past Appointment List. 2. Patient click on the past appointment that he/she want to follow up. 3. The details of the follow up appointment will be displayed together with options to choose the service, date and time. 4. Patient selects date and time for the follow up appointment. 5. Update Database. 6. Notification Management System sends a notification to the user about the upcoming appointment. 7. Notification Management System sends an email and/or SMS to the user about the upcoming appointment depending on his preferred mode of notification.
Alternative Flows:	
Exceptions:	2.3.EX.1 User does not select an appointment date: System requests the user to select a date. 2.3.EX.2 User does not select the appointment time: System requests the user to select a time. 2.3.EX.3 User selects a past date or a date less than 24 hours after the current date: System requests the user to book at least 24 hours in advance.
Includes:	View Past Appointment List, Update Database
Special Requirements:	
Assumptions:	3. The patient had entered his correct email address and contact number during registration. 4. The patient will complete any pre-appointment tests or actions before the appointment.
Notes and Issues:	

5.3.5. Use Case Description – View Past Appointment List

Use Case ID:	2.4		
Use Case Name:	View Past Appointment List		
Created By:	Deka	Last Updated By:	Ankur
Date Created:	February 10, 2015	Date Last Updated:	April 18, 2015
Actor:	Patient, Database Management System		
Description:	Viewing completed appointments list of patient		
Preconditions:	1. Patient has an account and has already logged in to the application. 2. Patient must already have one or more appointments that he has already attended.		
Postconditions:	Patients will see the list of recent appointments that the patient have done.		
Priority:	2		

Frequency of Use:	Once in two days
Flow of Events:	1. Retrieve From Database. 2. Patient will see all his/her past appointments within last 90 days.
Alternative Flows:	2.4.AC.1 If the patient is a new user, or have not gone to the appointments, he/she will not see the list of recent appointments.
Exceptions:	
Includes:	Retrieve From Database
Special Requirements:	
Assumptions:	
Notes and Issues:	

5.3.6. Use Case Description – Edit Appointment

Use Case ID:	2.5		
Use Case Name:	Edit Appointment		
Created By:	Ankur	Last Updated By:	Shreyas
Date Created:	February 10, 2015	Date Last Updated:	April 18, 2015
Actor:	Patient, Database Management System, Notification Management System		
Description:	Editing/Changing appointment		
Preconditions:	1. Patient has an account and has already logged in to the application 2. Patient already has an existing appointment. 3. The appointment is not a past or an ongoing appointment.		
Postconditions:	Patient has been notified about his upcoming appointment.		
Priority:	2		
Frequency of Use:	Twice a month		
Flow of Events:	1. Patient chooses to edit an appointment. 2. Retrieve From Database. 3. Patient makes the necessary changes in the form for editing appointment and submits those details. 4. Update Database. 5. Notification Management System sends a notification to the user about the edited appointment. 6. Notification Management System sends an email and/or SMS to the user about the edited appointment depending on his preferred mode of notification. 7. The patient is redirected back to the home page.		
Alternative Flows:	2.5.AC.1 If patient has follow-up actions, Patient will be asked to change their appointments for the following appointments if it is necessary, some treatments may require buffer time before the next follow-up appointment therefore the date and time for the follow up actions may also need to be changed.		
Exceptions:	2.5.AC.1 Patient is not allowed to change an appointment that needs to be after certain actions to an earlier time.		
Includes:	Update database, Retrieve From Database		
Special Requirements:	Patient can only change his/her appointment 4 hours before the real appointment, otherwise the system will disable the option to make changes		

	to the appointment.
Assumptions:	1. The patient had entered his correct email address and contact number during registration. 2. The patient will complete any pre-appointment tests or actions before the appointment.
Notes and Issues:	

5.3.7. Use Case Description – View Appointment

Use Case ID:	2.6		
Use Case Name:	View Appointment		
Created By:	Deka	Last Updated By:	Ankur
Date Created:	February 10, 2015	Date Last Updated:	April 18, 2015
Actor:	Patient, Database Management System		
Description:	Viewing upcoming appointments for patients, view appointment		
Preconditions:	1. Patient has an account and has already logged in to the application 2. Patient already has an existing appointment.		
Postconditions:	Patients will see the detailed information about their appointment		
Priority:	1		
Frequency of Use:	Once in two days		
Flow of Events:	1. Patient that has logged in will be directed to the main page which display the list of appointments 2. Patient click on one of the appointments to view its details 3. Retrieve From Database. 4. The appointment details will be displayed		
Alternative Flows:			
Exceptions:	2.6.EX.1 If a new patient is a new user, there will be no existing appointment available to see.		
Includes:	Retrieve from Database		
Special Requirements:	1. Patient must be logged in 2. Patient must already have an appointment.		
Assumptions:			
Notes and Issues:			

5.3.8. Use Case Description – Delete Appointment

Use Case ID:	2.7		
Use Case Name:	Delete Appointment		
Created By:	Ankur	Last Updated By:	Ankur
Date Created:	Feb 10, 2015	Date Last Updated:	April 18, 2015
Actor:	Patient, Database Management System		
Description:			
Preconditions:	1. Patient has an account and has already logged in to the application 2. Patient already has an existing appointment.		

	3. The appointment is not a past or an ongoing appointment.
Postconditions:	
Priority:	2
Frequency of Use:	Twice a month
Flow of Events:	1. User choose the option to manage appointments from the menu. 2. According to which display the user preferred, system displays the appointments that the user currently had either in calendar form or list form. 3. User choose to cancel an appointment. 4. System will confirm that the cancellation is possible and verify with the user. 5. Update Database. 6. System updates the user about the cancellation appointment result and any notifications regarding the upcoming appointment will be deleted.
Alternative Flows:	
Exceptions:	
Includes:	Update database
Special Requirements:	Cancellation should be done the latest is four hours before the appointment
Assumptions:	
Notes and Issues:	

5.3.9. Use Case Description – View Appointment List

Use Case ID:	2.8		
Use Case Name:	View Appointment List		
Created By:	Deka	Last Updated By:	Ankur
Date Created:	February 10, 2015	Date Last Updated:	April 18, 2015
Actor:	Patient, Database Management System		
Description:	Viewing upcoming, past and ongoing appointments for patients		
Preconditions:	1. Patient has an account and has already logged in to the application		
Postconditions:	Patients will see the list of appointments that they have, including the past, ongoing and upcoming appointments		
Priority:	1		
Frequency of Use:	Once every two days		
Flow of Events:	1. Patient logs in. 2. Retrieve From Database. 3. Patient will see all his/her appointments		
Alternative Flows:	1. If the patient is a new user, he/she will not see list of appointments		
Exceptions:			
Includes:	Retrieve From Database		
Special Requirements:			
Assumptions:			
Notes and Issues:			

5.4. Medical History Subsystem

5.4.1. Use Case Model

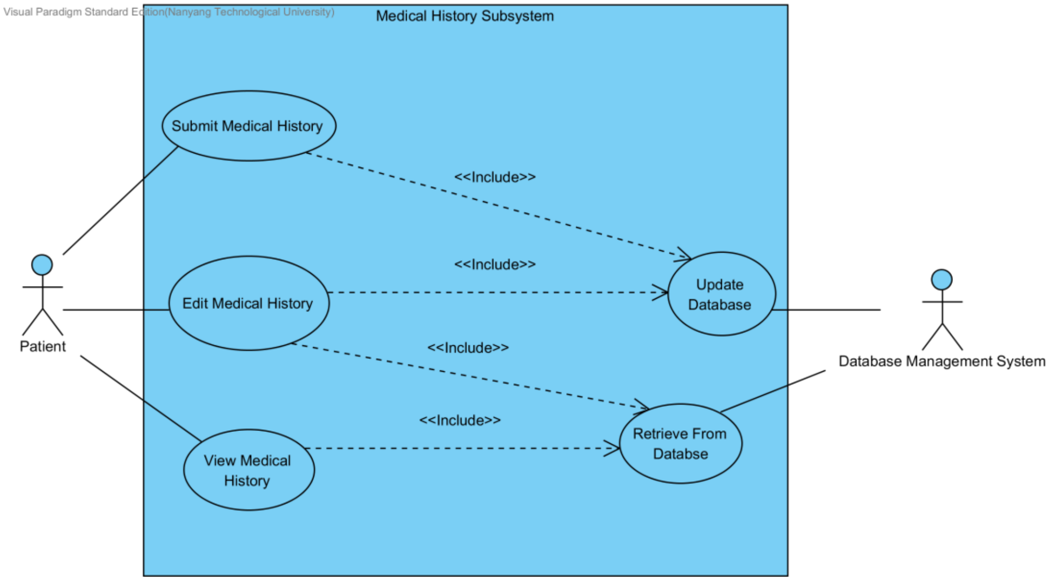

Figure 5

5.4.2. Use Case Description – Submit Medical History

Use Case ID:	3.1		
Use Case Name:	Submit Medical History		
Created By:	Zillion Govin	Last Updated By:	Ankur
Date Created:	February 12, 2015	Date Last Updated:	April 17, 2015
Actor:	Patient, Database Management System		
Description:	Fill in medical history form		
Preconditions:	Patient has signed up for an account		
Postconditions:	New patient account is created		
Priority:	2		
Frequency of Use:	Every time a user sign up.		
Flow of Events:	1. Patient finishes the sign up phase. 2. System creates an account for the user and displays the medical history form. 3. Patient fills in the medical history form and submits it. 4. Update Database. 5. System redirects user back to the login page.		
Alternative Flows:	3.1.AC.1.1 User can choose not to fill in the medical form after system displays the form by pressing back. 3.1.AC.1.2 System asks the user to confirm his choice using alert dialog. 3.1.AC.1.2.1 If user agrees, system redirects user back to login,		

	else system will stay on displaying the form.
Exceptions:	
Includes:	Update Database.
Special Requirements:	
Assumptions:	
Notes and Issues:	

5.4.3. Use Case Description – Edit Medical History

Use Case ID:	3.2		
Use Case Name:	Edit Medical History		
Created By:	Zillion Govin	Last Updated By:	Ankur
Date Created:	February 12, 2015	Date Last Updated:	April 18, 2015
Actor:	Patient, Database Management System		
Description:	Edit Medical History of Patient		
Preconditions:	Patient must be logged in		
Postconditions:			
Priority:	2		
Frequency of Use:	Depends on Patient (Recommended once in 3 months and after an appointment).		
Flow of Events:	1. Patient selects the 'Medical History' to view the medical history inputted earlier. 2. Patient selects to Edit the 'Medical History'. 3. Retrieve From Database. 4. System displays the medical history form, with previous information inputted by user. 5. Patient makes necessary changes and submit it. 6. Update Database. 7. System redirects user back to Homepage.		
Alternative Flows:	3.2.AC.6 If user presses back after selecting edit, system will simply direct user to the Homepage without updating database.		
Exceptions:			
Includes:	Retrieve From Database, Update Database		
Special Requirements:			
Assumptions:			
Notes and Issues:			

5.4.4. Use Case Description – View Medical History

Use Case ID:	3.3		
Use Case Name:	View Medical History		
Created By:	Zillion Govin	Last Updated By:	Ankur
Date Created:	February 17, 2015	Date Last Updated:	April 17, 2015
Actor:	Patient, Database Management System		
Description:	View Medical History of Patient		
Preconditions:	Patient is logged in		

Postconditions:	
Priority:	3
Frequency of Use:	Patient's Discretion
Flow of Events:	1. Patient selects to view Medical History. 2. Retrieve From Database. 3. System displays the medical history details, ongoing treatments and medications of the patient.
Alternative Flows:	
Exceptions:	
Includes:	Retrieve From Database
Special Requirements:	
Assumptions:	
Notes and Issues:	

5.5. Notification Subsystem

5.5.1. Use Case Model

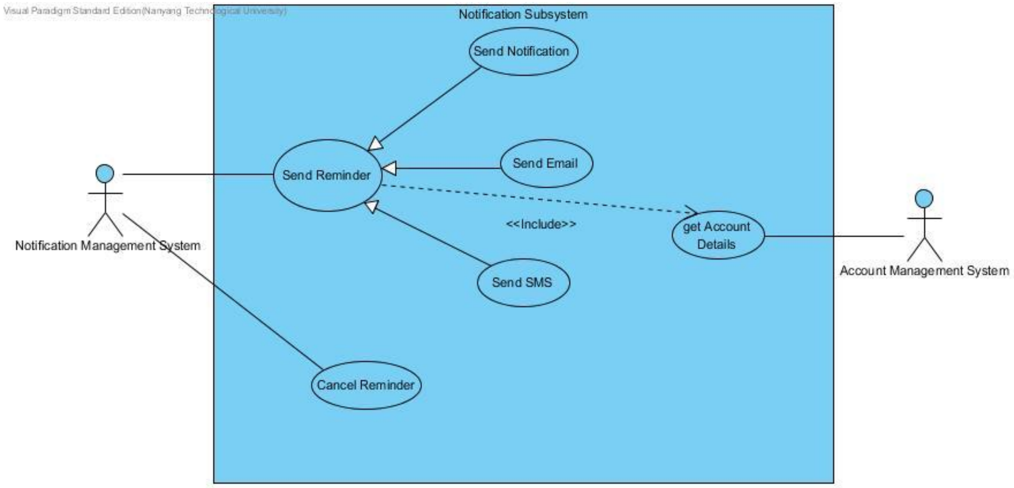
Visual Paradigm Standard Edition(Nanyang Technological University)
Notification Subsystem
Send Notification
Send Reminder
Send Email
<<Include>>
get Account Details
Send SMS
Cancel Reminder
Notification Management System
Account Management System

Figure 6

5.5.2. Use Case Description – Send Notification

Use Case ID:	4.1		
Use Case Name:	Send Notification		
Created By:	Ankur	Last Updated By:	Ankur
Date Created:	February 10, 2015	Date Last Updated:	April 17, 2015
Actor:	Notification management system, Account management system		
Description:	Send push notifications		
Preconditions:	1. Patient must have an account. 2. Patient must be logged in. 3. An appointment must exist.		
Postconditions:			

Priority:	2
Frequency of Use:	24 hours before an upcoming appointment.
Flow of Events:	1. An alarm is set by the Notification Management System. 2. Notification Management System generates a push notification 24 hours before the appointment time.
Alternative Flows:	
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

5.5.3. Use Case Description – Send Email

Use Case ID:	4.2		
Use Case Name:	Send Email		
Created By:	Ankur	Last Updated By:	Ankur
Date Created:	February 10, 2015	Date Last Updated:	April 17, 2015
Actor:	Notification management system, Account management system		
Description:	Send an email		
Preconditions:	1. An upcoming appointment in next 24 hours or the patient has forgotten his password.		
Postconditions:			
Priority:	2		
Frequency of Use:	Everytime there is an appointment due in the next 24 hours or the patient wants to retrieve his password.		
Flow of Events:	1. Get Account Details. 2. Send an email to the email address obtained from the Account Details. (Send a reminder for the appointment, sends password to the email if it's a password retrieval.)		
Alternative Flows:			
Exceptions:			
Includes:	Get Account Details		
Special Requirements:			
Assumptions:			
Notes and Issues:			

5.5.4. Use Case Description – Send SMS

Use Case ID:	4.3		
Use Case Name:	Send SMS		
Created By:	Ankur	Last Updated By:	Ankur
Date Created:	February 10, 2015	Date Last Updated:	April 17, 2015
Actor:	Notification management system, Account management system		
Description:	Send SMS reminder		
Preconditions:	1. An upcoming appointment in next 24 hours.		
Postconditions:			

Priority:	2
Frequency of Use:	Everytime there is an appointment due in the next 24 hours.
Flow of Events:	1. Get Account Details. 2. Send a message to the phone number obtained from the Account Details.
Alternative Flows:	
Exceptions:	
Includes:	Get Account Details
Special Requirements:	
Assumptions:	
Notes and Issues:	

5.5.5. Use Case Description – Cancel Reminder

Use Case ID:	4.4		
Use Case Name:	Cancel Reminder		
Created By:	Ankur	Last Updated By:	Ankur
Date Created:	February 10, 2015	Date Last Updated:	April 17, 2015
Actor:	Notification management system, Account management system		
Description:	Cancel Reminder		
Preconditions:	1. An existing appointment must be cancelled.		
Postconditions:	1. Reminder is cancelled.		
Priority:	3		
Frequency of Use:	Everytime an appointment is cancelled.		
Flow of Events:	1. Alarm for the original appointment is cancelled. 2. The notification reminder is not sent.		
Alternative Flows:			
Exceptions:			
Includes:			
Special Requirements:			
Assumptions:			
Notes and Issues:			

5.5.6. Use Case Description – Get Account Details

Use Case ID:	4.5		
Use Case Name:	Get Account Details		
Created By:	Ankur	Last Updated By:	Ankur
Date Created:	February 10, 2015	Date Last Updated:	April 17, 2015
Actor:	Account management system		
Description:	Retrieve Account Details		
Preconditions:	1. An account must exist. 2. The patient must be logged in.		
Postconditions:			
Priority:	2		
Frequency of Use:	High		
Flow of Events:	1. Account Management System retrieves the account details of the logged in patient.		

	2. The requested details are shared with the calling functionality.
Alternative Flows:	
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

5.6. Database Subsystem

5.6.1. Use Case Description – Retrieve From Database

Use Case ID:	5.1		
Use Case Name:	Retrieve From Database		
Created By:	Ankur	Last Updated By:	Ankur
Date Created:	February 12, 2015	Date Last Updated:	April 18, 2015
Actor:	Database Management System		
Description:	Retrieving information from the database		
Preconditions:			
Postconditions:			
Priority:	1		
Frequency of Use:	Everyday		
Flow of Events:	1. Database Management System retrieves the requested information from the database. 2. The information is shared with the calling function or activity.		
Alternative Flows:			
Exceptions:			
Includes:			
Special Requirements:			
Assumptions:			
Notes and Issues:			

5.6.2. Use Case Description – Update Database

Use Case ID:	5.2		
Use Case Name:	Update Database		
Created By:	Ankur	Last Updated By:	Ankur
Date Created:	February 12, 2015	Date Last Updated:	April 18, 2015
Actor:	Database Management System		
Description:	Updating database with new or changed information		
Preconditions:			
Postconditions:	Relations in database will be updates with new information.		
Priority:	1		
Frequency of Use:	Everyday		

Flow of Events:	1. Database Management System retrieves the information inputted or changed by the patient. 2. Database Management System updates the database with the new or changed information.
Alternative Flows:	
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

6. Glossary

Term	**Definition**
User	The person who use the mobile application
Patient	A user who request appointment, a person receiving or registered to receive medical treatment.
Appointment	A period of time booked by the patient for their medical treatment.
NRIC	Unique Identification Number of a person. Generally, NRIC is issued by the country where the person resides.
Preferred Notification Mode	The preferred mode of notification for receiving reminders and updates related to the appointment from the medical facility.
Username	A unique identifier for each user to log in to the application.
Password	A way to verify the existence of a user account in the application database.
Referral	A reference from a doctor or clinic regarding the continuation of the treatment according to the diagnosis.
Follow-Up	A follow up a previous appointment so that the doctor is already aware of the diagnosis and continues with the treatment as required.
Clinic Location	A medical centre or building established by the medical facility in a particular country or region.
Specialty	A department within the clinic which deals with certain types of treatments.
Service	A specific type of service provided under a certain specialty or department.
Doctor	A person who provides medical treatment to patients.
ENT	A department in clinics which deals with medical problems related to Ear, Nose and Throat.
Dental Services	A department in clinics which deals with medical problems related to teeth and gums.
Women’s Health	A department in clinics which deals with medical problems to women.
General Medicine	A department in clinics which deals with prevention, diagnosis and treatment of general adult diseases.

Pre Appointment Actions	Some requirements that patients' need to fulfil before their appointment with the doctor.

Specialty Names	**Service Names**	**Definition**
ENT	General	All conditions and disorders of the head and neck.
	Periodic ENT	All periodic medical conditions of ear, nose and throat such as sinus diseases.
	OSA	Obstructive Sleep Apnoea
	Otology	All medical conditions related to hearing and related sensory functions.
Dental Services	Routine Scaling	Cleaning of teeth by dentist.
	Polishing	Removal of plaque and whitening of teeth.
	Fillings	Filling in infected or damaged premolars or molar teeth with artificial materials.
	Tooth Extraction	Removal of tooth from it socket in the bone.
	Root Canal	Sequence of treatment for infected pulp of tooth to eliminate infection and protect decontaminated tooth from future infection.
Women's Health	Gynaecologists	Treating medical conditions related to female reproductive system and the breasts.
	Obstetrician	Management of pregnancy, labor and birth related treatments.
General Medicine	Dietetic Services	Dietetic counselling and nutritional intervention to prevent risk of diet related diseases.
	Physiotherapy	Concerned with enhancing mobility for physically affected parts of body through therapy.
	Child Care	Deals with prevention, diagnosis and treatment of diseases common to small children.
	Chronic Care	Addresses pre-existing or long term illness.

7. References

Provide a list of all documents and other sources of information referenced in the SRS and utilized in developing the SRS. Include for each the document number, title, date and author.

Document No.	Document Title	Date	Author
1.	CZ2006 Lecture Notes		Graham Leedham Xing Zhenchang
2.	SRS Template		CZ2006 Lab Material
3.	Use Case Description Template		Karl E. Wiegers

8. Revision History

Identify changes to the SRS.

Version	Date	Name	Description
1.0	Feb 5, 2015	Deka	Initial Specification of Requirements
1.0	Feb 8, 2105	Shreyas	Adding created Use Case Descriptions and more updates related to project description
1.0	Feb 15, 2015	Ankur	Adding created Use Case Descriptions and updating Functional, Non Functional Requirements and Interface Requirements. Creating Glossary
2.0	April 14, 2015	Shreyas	Finalising Use Case Descriptions
2.0	April 17, 2015	Ankur	Finalising Use Case Descriptions, Finalising Requirements, Finalising Glossary, Final Formatting of the document