Product Requirements

Team: Stephen Cook, Chris Letourneau, Oren Rosin, Xixiao Yue, Georgiy Rozenshteyn

Revision Number	*Revision Date*	*Summary of Changes*	*Author(s)*
0.1	*02/04/2017*	*Creation of initial requirement document*	*Stephen Cook*
0.2	*02/04/2017*	*Addition of Use Case Descriptions for Viewing of Activity Log and System Statistics*	*Oren Rosin*
0.3	*02/05/2017*	*Addition of Use Case Descriptions for Appointments*	*Stephen Cook*
0.4	02/06/2017	*Addition of Use Case Descriptions for Updating Patient Information*	*Chris Letourneau*
0.5	02/06/2017	*Addition of Use Case Descriptions for Logging System Activity*	*Georgiy Rozenshteyn*
0.6	02/06/2017	*Addition of Use Case Descriptions for Adding Prescriptions and Viewing Medical Info*	*Xixiao Yue*
0.7	02/07/2017	*Finalizing all requirements and formating*	*Stephen Cook*
0.8	02/09/2017	*Requirements Revision*	*Stephen Cook*
0.9	02/15/2017	*Updated the Usecase Diagram*	*Stephen Cook*
1.0	03/07/2017	*Finalized Requirements for Release 1*	*Georgiy Rozenshteyn*
1.1	03/21/2017	*Added new requirement for Release 2*	*Oren Rosin*
1.2	03/25/2017	*Update Requirements for R2 planning*	*Whole Team*
1.3	03/27/2017	*Update Use Case Context Diagram*	*Georgiy Rozenshteyn*
2.0	05/05/2017	*Update Requirements for R2*	*Georgiy Rozenshteyn*
2.1	05/07/2017	*Update Use Case Context Diagram (Readability)*	*Georgiy Rozenshteyn*
2.2	05/08/2017	*Finalize Requirements for Release 2*	*Whole Team*

Brief problem statement

We represent a funding group (HAccelerator) chartered to create applications for the benefit of health-care across the country. The project we currently want to make a reality will be called **HealthNet**. At its core, HealthNet is meant to enable their hospitals in the US to be able to manage both employees and patients. The successful implementation should make it easy for users to effortlessly sign-up as patients so that the hospital can, without difficulty, manage their procedures and patient related tasks to optimize day-to-day work-flow.

The HealthNet product is intended to improve hospitals by providing an easy mechanism for managing employees, gathering statistical data on the inner workings of the hospital, signing up patients, making appointments, and allowing ease of transfer of both patients and their information between hospitals.

We want a product whose emphasis is on ease of use, whose navigation is straightforward and where the status of any, and all, information shown is clearly displayed. Ultimately, a system where understanding and communication about hospital and patient matters is improved.

HealthNet is a web application project intended to improve communication and ease of access of data by healthcare providers. HealthNet is geared towards enhancing the management of patient and employee-related information, as well as facilitation of day-to-day hospital business by focusing on ease of use and navigation. Special consideration will be given to the development of an intuitive front-end system allowing for straightforward transfer of data and providing a platform for effective communication between patients and healthcare professionals.

Stakeholders

HAccelerator Board of Directors – oversee the projects funding and expenses. Have vested interest in the proven success of the product but are not involved in the planning and execution.
HAccelerator Product Owner – will act as principle representative for HealthNet product needs. He/she champions the product with the Board of Directors, helps facilitate product decisions and has the ultimate say on when and what features should be released.
Software Engineering Team – is responsible for the day-to-day operations and coordination of all aspects related to the software product's life-cycle. This include, among others: planning and delegation of team roles and responsibilities; elicitation and clarification of requirements; analysis and design; implementation, testing and release of all software components.
Beta Testing Team – represent the target user base for HealthNet. Will be available in later phases of the project to conduct acceptance testing and provide feedback on product release.

Users profile

The target user (patient) must:

- Have basic experience using computers and browsing the internet. Has filled out online forms or surveys and may have purchased or sold a product.
- Have a computer with access to the internet
- Have an interest in improving their health by using an online way of interacting with their hospital
- Be willing to share information such as home address and contact information as well as more personal information such as medical history

Doctor:

- Have moderate experience using computers and browsing the internet. Has filled out online forms or surveys and may have purchased or sold a product.
- Have a computer with internet access
- Basic cyber security knowledge
- Ability to upload and update necessary information pertaining to patient health.
- Convey availability and office hours

Nurse:

- Have moderate experience using computers and browsing the internet. Has filled out online forms or surveys and may have purchased or sold a product.
- Basic cyber security knowledge
- Have a computer with internet access
- Ability to update necessary information pertaining to patient health.

Admin:

- Have advanced experience using computers and browsing the internet.
- Have a computer with internet access
- Advance cyber security knowledge
- Familiarity with database management.
- Knowledge of Django admin tools

System requirements

At a high-level this project will be source controlled in SVN, run on Django using python, sqlite and needs to be compatible with the latest browsers.

At a lower, more granular, level, this application is expected to run on Django version 1.9.1 and Python version 3+ - specific version does not make any difference, so long as it is 3 or above (3.4.3 was used during development). The version control system is SVN, which is interacted with via TortoiseSVN, an Apache SVN client, implemented through a Windows shell extension. The application will be produced on a Windows OS environment, however it should function the same

in a Unix or Macintosh environment, provided all other system requirements are accounted for. The application is architecture independent, thus both 32 and 64-bit architectures will provide the same functionality. The application will interact with an sqlite (sqlite3) database

Feature requirements (user stories)

The following list of user stories is neither final nor comprehensive. You must consider it your responsibility to maintain its relevance, clarify any misunderstandings and keep it up-to-date. Any changes must be discussed with the Product Owner for approval.

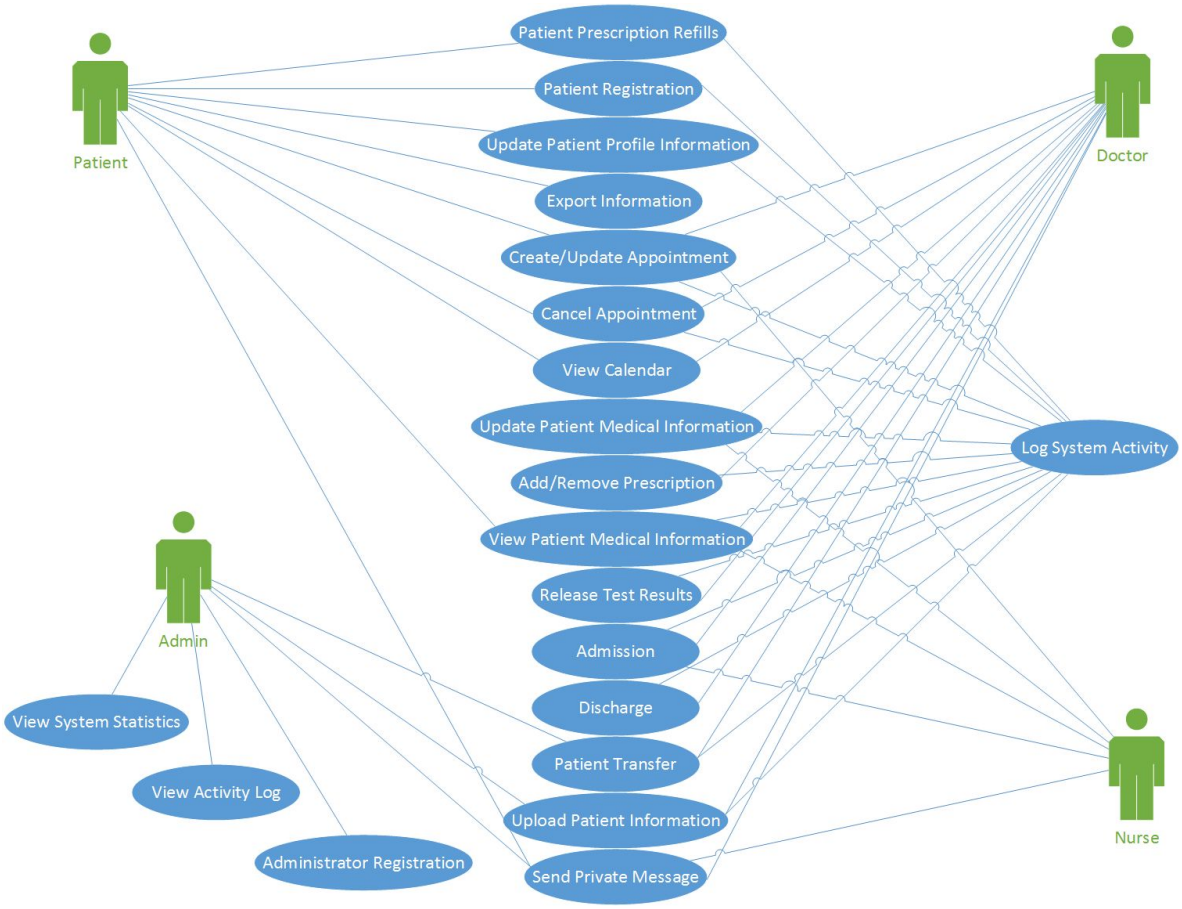
No.	User Story Name	Description	Release
1	Patient Registration	Users sign up to become a Patient by providing their personal contact information, proof of insurance and unique login credentials. Additionally, a patient should provide the system with some basic medical profile information, a choice of preferred hospital and emergency contact information (linked to another patient if they are already in the system).	**R1**
2	Administrator Registration	Doctors, Nurses, and Administrators will be added to the system by other administrators. All information for creating these new accounts will be done through an administrator account.	**R1**
3	Update Patient Profile Information	Patients can update their profile information.	**R1**
4	Update Patient Medical Information	Doctors and Nurses can update patient medical information.	**R1**
5	Export Information	Patients will be able to export their information and their test results from the system with relevant privacy warnings.	**R2**
6	Create or Update Patient Appointment	Patients, doctors and nurses can create or update an appointment with a doctor and at one of the doctor's available locations.	**R1**

		If the patient or doctor already has an appointment at the time selected, then the system will not allow for the appointment.	
7	Cancel Patient Appointment	Patients can cancel their existing appointments. Doctors can cancel their existing appointments. Nurses cannot cancel (only modify) existing appointments.	**R1**
8	Appointment Calendar	Doctors and patients will easily be able to view all of their appointments in a calendar view. Nurses will be able to see all appointments for the day and week between Patients and Doctors.	**R1**
9	Add/Remove Prescriptions	Doctors can add or remove a prescription to a patient record. Nurses can view the prescriptions of patients belonging to the same hospital. Patients can view their prescriptions from their account.	**R2**
10	Viewing Patient Medical Information, Prescriptions and Tests and Results	Doctors can view all medical information for any patient in the system (regardless of Hospital). Nurses can only view patient medical information in the hospital they work for. Patients can view their tests (pending or completed) and view the corresponding results for those tests that have been released by the doctor. Prescriptions and other non-sensitive information is viewable by the patient without a need for doctor's release.	**R1**
11	Release Test Results	Doctors (within the patient's hospital) can, upon evaluating a patient's test results, release them for view by that patient.	**R2**

		Comments may be added to the specific test result for view by the patient.	
12	Logging System Activity	For security, many actions in the system will be logged for review at a later date. Some examples of actions to be logged include but are not limited to updating of a Patient's information, viewing of a Patient's information/records, and transfers of a Patient from one hospital to another.	**R1**
13	Admission and Discharge to/from Hospital	Doctors and Nurses can admit a patient to the hospital for an extended stay (reasons could be: emergency, observation, surgery, etc.). These are typically unexpected visits but can result from a decision made after a scheduled appointment. This event is recorded by the system. Doctors are the only ones to approve a patient's discharge from the Hospital. This event is recorded by the system.	**R2**
14	Viewing Activity Log	Administrators will be able to view the logs of all system activity for a given time-frame at their hospital. Some examples of this might be: - breakdown of the viewing activity of patient records or by system user - most common system activities (or by user) Other important and informative statistics yet to be determined.	**R1**
15	Viewing System Statistics	Administrators will be able to view compiled statistics for a given time-frame at their hospital. Some examples of this might be: - number of patients visiting the hospital - average number of visits per patient - average length of stay (from admission to discharge) - most common reasons for being admitted to the	**R2**

		hospital - prescription statistics Other important and informative statistics yet to be determined.	
16	Patient Transfer	Patient can be transferred between hospitals. Transfers can be carried out by either administrators or by doctors (ones who are at the receiving hospital).	**R2**
17	Upload Patient Information	Doctors will be able to upload the results of a patient's tests if needed. Doctors will be able to upload images such as those used in X-Rays to update a patient's record. Uploads are considered as updates to a patient's medical information.	**R2**
18	Send Private Message	Doctors, nurses, patients and administrators can send private messages of limited length via the system.	**R2**
19	Patient Prescription Refills	Patients may order prescription refills through the system. The prescription must have been prescribed by a doctor, contain dosage information, and not be expired. There must be insurance coverage. The system will provide an availability date and cost. Payment is made at the time of pickup independent of HealthNet.	**R2**

Use case context diagram


Patient
Patient Prescription Refills
Patient Registration
Update Patient Profile Information
Export Information
Create/Update Appointment
Cancel Appointment
View Calendar
Update Patient Medical Information
Add/Remove Prescription
View Patient Medical Information
Release Test Results
Admission
Discharge
Patient Transfer
Upload Patient Information
Send Private Message
Doctor
Log System Activity
Nurse
Admin
View System Statistics
View Activity Log
Administrator Registration

Use case description

Use Case Number:	UC-01
Use Case Name:	Patient Registration
Overview:	Registrant shall provide personal, medical, and emergency contact information to the System upon registering and becoming a Patient.
Actor(s):	Registrant
Pre-condition(s):	- System has been setup and configured. - System is running and open for registrations. - Registrant has accessed website via URL
Scenario Flow:	Main (success) Flow: 1. Registrant selects option to register 2. System requests personal information 3. Registrant provided personal information. 4. System verifies required information is provided. ● If information is invalid System displays message. Return to Step 2 5. System requests basic medical information 6. Registrant provides medical information 7. System verifies required information is provided. ● If information is invalid System displays message. Return to Step 5 8. System requests emergency contact information 9. Registrant provides emergency contact information 10. System verifies required information is provided ● If information is invalid System displays message. Return to Step 8 11. System requests login information 12. Registrant provides login information 13. System verifies required information is provided ● If information is invalid System displays message. Return to Step 11 14. System displays confirmation of registration

Alternate Flows:	Alternate Flow #1: After Step 2 in success scenario System will display the option to Cancel the registration process. The following steps would occur: 1. Registrant selects option to cancel during registration 2. System requests confirmation to cancel 3. Registrant confirms intent 4. System returns to main screen Alternate Flow #2: The emergency contact information is an existing user in the system. After step 10 the following steps would occur: 1. Registrant selects option to select an emergency contact from the system 2. System displays a search bar for the Registrant to input the user's name 3. Registrant inputs the user's name and presses enter 4. System returns a list of users with matching names 5. Registrant chooses intended user 6. System sets that user as an emergency contact
Post Condition:	If registrant did not complete registration. System does not store Registrant's information. If registrant complete registration store information.

Use Case Number:	UC-02
Use Case Name:	Administrator Registration
Overview:	Admin Registration requires a nurse, doctor, or admin to give all required information to create a HealthNet Account. Once information is gathered, an Admin creates an account.
Actor(s):	Admin, Doctor, Nurse
Pre-condition(s):	- System has been setup and configured. - System is running and open for registrations. - Registrant has access to Admin.
Scenario Flow:	Main (success) Flow: 1. Registrant contacts admin.

	2. Admin gets prefered login information. 3. System request account type. 4. Admin selects account type 5. System request login information 6. Admin gives login information 7. System checks validity ● If information is invalid, Display error message and return to step 3 8. System displays confirmation.
Alternate Flows:	Alternate Flows #1: If registrant is a doctor after step 7 doctor will add personal account information : 1. Doctor logins into account 2. System request Name, Hospital, Office Hours, Contact Info and Speciality 3. Doctor inputs information 4. System checks validity ● If information is invalid, Display error message and return to step 2 5. System saves changes. Alternate Flows #2: If registrant is a nurse after step 7 nurse will add personal account information : 1. Nurse logins into account 2. System request Name and Hospital. 3. Nurse inputs information 4. System checks validity ● If information is invalid, Display error message and return to step 6. System saves changes.
Post Condition:	If registrant did not complete registration, system does not store registrant's information. If registrant completes registration, store information.

Use Case Number:	UC-03
Use Case Name:	Update Patient Profile Information

Overview:	Allows the patients to update their profile information, such as contact information.
Actor(s):	Patient
Pre-condition(s):	● System had been set up and is operational ● Patient has created an account ● Patient has access to the website via URL
Scenario Flow:	Main (success) Flow: 1. Patient logs into their Healthnet account. 2. Patient then clicks on their profile information to view it. 3. Patient then clicks on the EDIT button. 4. Patient can then change any of their profile information. 5. To finalize the changes the Patient must click the SAVE button. ● If the information entered is invalid. Display error message, and return to step 4 6. The system saves the changes to the patient's profile information.
Alternate Flows:	Alternate Flows: After step 4, if patient leaves without pressing the SAVE button. 1. System warns Patient that the changes were not saved and asks if they want to continue leaving. 2. If the Patient clicks YES System does not save changes. ● If the Patient clicks NO, continue to step 5 of Main Flow.
Post Condition:	The Patient's profile information has been updated. If the Patient left at step 4, the profile information is unchanged.

Use Case Number:	UC-04
Use Case Name:	Update Patient Medical Information
Overview:	Allows Doctors and Nurses to update a Patient's medical information
Actor(s):	Doctor and Nurse
Pre-condition(s):	● System had been set up and is operational. ● Patient has created an account. ● Doctor/Nurse has access to an account.

Scenario Flow:	Main (success) Flow: 1. Doctor/Nurse logs onto their account. 2. They then find the patient through the search. 3. They then click the UPDATE MEDICAL INFORMATION button. 4. The Doctor/Nurse can then edit the medical information. 5. To finalize the changes, the Doctor/Nurse must click the SAVE changes. ● If any of the information is invalid, the System displays an error, and return to step 4. 6. The Doctor/Nurse re-enter their login information to confirm the changes. ● If the login information in invalid, the System displays and error and return to step 6. 7. System saves changes to the medical information.
Alternate Flows:	Alternate Flow #1: If at step 1, patient is attempting to update medical information rather than doctor/nurse 1. Medical information is not liable to be updated by patient. 2. Use Case not applicable, patient can only view limited medical information, cannot edit 3. Medical information remains unchanged
Post Condition:	The Patient's medical information is updated.

Use Case Number:	UC-05
Use Case Name:	Export Patient Information
Overview:	Allows a patient to Export his information
Actor(s):	Patient
Pre-condition(s):	● User is signed into a patient account ● Patient has a valid and active account
Scenario Flow:	Main (Success) Flow: 1. Patient is logged in and viewing medical information. 2. Patient clicks the export button. 3. System requires a password validation to continue 4. Patient inputs password 5. System validates password 6. System displays privacy warning 7. Patient confirms indent to continue 8. System exports tests and medical information to user
Alternate Flows:	Alternative Flow #1: Occurs after step 5 of main flow if password is incorrect

	1. System indicated invalid password and returns to step 3 Alternative Flow #2: Occurs at step 7 if patient doesn't wish to continue 1. System returns to test result page.
Post Condition:	Patient downloads medical information and Healthnet returns to tests page.

Use Case Number:	UC-06
Use Case Name:	Create or Update Appointment
Overview:	Allows a patient, doctor, or nurse to create an appointment with a doctor.
Actor(s):	Patient, Doctor, Nurse
Pre-condition(s):	● System has been set up and accepts appointments ● User has access to appointment calendar ○ User has an existing appointment to update ● User is logged into their account
Scenario Flow:	1. User opens appointment calendar 2. User selects option to create an appointment 3. System requests doctor's information ● Information includes name, hospital, and speciality. 4. User inputs doctor's information 5. System retrieves the doctor's appointment schedule ● If information can not be found. System prints out an error message and returns to step 3. 6. System request user to confirm correct doctor. 7. System request Date and Time 8. User inputs date and time 9. System verifies date and time ● If time isn't available, print error message and return to step 4 10. System confirms intent to create an appointment ● System displays all information about appointment: date, time, location, doctor. 11. User confirms appointment 12. System creates appointment and adds to calendar 13. System sends update to other Calendars (Doctor's and Hospital's Nurses) 14. System locks-in time for the doctor so other appointments are not created at same time.
Alternate Flows:	Alternative Flow #1: After opening calendar, User selects an already existing appointment. Starts at step 2. 1. System requests confirmation of changing appointment date

	● If user cancels update, System does nothing to existing appointment 2. User confirms desire to change appointment 3. System requests new date and time 4. User inputs new date and time 5. System verifies time is available ● If time not available, System prints an error message and returns to step 3 6. System confirms appointment information ● Displays doctor, date, time and location 7. User confirms new appointment information. ● If information is incorrect, User hits no, System returns to step 2. ● If user wants to cancel update, hits cancel i. System then cancels the update and returns to Calendar 8. System Updates appointment and updates calendar Alternative Flow #2: After step 2 in Main Flow the option to cancel an appointment will be available. If cancel is selected the following steps occur: 1. User selects option to cancel appointment creation 2. System request confirmation to cancel 3. User confirms intent 4. System returns to calendar Alternative Flow #3 : After step 10, when System requires confirmation, If user does not confirm intent to create or cancels 1. System ends creation process. 2. System alerts user of cancellation process. 3. System returns to calendar
Post Condition:	If an appointment is created or updated, The system updates the calendar to add the new appointment. If canceled, System doesn't add appointment and returns to calendar

Use Case Number:	UC-07
Use Case Name:	Cancel Patient Appointment
Overview:	Doctors and Patients are given the ability to cancel existing appointments
Actor(s):	Doctor, Patients
Pre-condition(s):	● System has been set up and accepts appointments ● User has access to appointment calendar ● User has an existing appointment to cancel ● User is logged into their account

Scenario Flow:	1. User selects an existing appointment on appointment calendar 2. User selects cancel appointment option 3. System confirms use intent 4. User confirms intent 5. System verifies appointment is able to be canceled ● If appointment can't be canceled, System tells user and returns to calendar 6. System cancels appointment. 7. System sends message and updates the Calendars of all involved.
Alternate Flows:	Alternative Flow #1: If user in set 4 does not confirm intent or denies cancellation, the following happens: 1. User does not confirm intent 2. System cancels cancellation process 3. System alerts user that process is canceled 4. System returns to Calendar without canceling appointment.
Post Condition:	If complete, appointment is canceled, return to Calendar

Use Case Number:	UC-08
Use Case Name:	Appointment Calendar
Overview:	Allows Doctors and Patients to see their appointments, Allows Nurses to see all appointment in a day and a week. Appointment edits and cancellations should be done here
Actor(s):	Doctor, Patient, Nurses
Pre-condition(s):	● System has been set up and accepts appointments ● User has access to appointment calendar ● User is logged into their account
Scenario Flow:	1. User selects view calendar 2. System gathers account information. Checks for Existing Appointments ● Alternative Flow 1 is for nurses 3. System gets appointment information of doctor or patient ● If user has no appointments, System displays a calendar with no appointments ● System also says no appointments scheduled 4. System displays appointment times, dates and locations for the week ● An alternative day view will be available 5. User selects day view option. 6. System checks for today's date. 7. System gets information for today's date and displays appointments.

	8. User select next day 9. System gets information for the next day 10. System displays appointments for next day
Alternate Flows:	Alternative Flow #1: If system detects user is nurse in step two, the following happens 1. Gather all appointment information for the hospital. 2. System displays information for the week. 3. System displays all appointments for the week on a calendar. 4. System checks for today's date. 5. System gets information for today's date and displays appointments. Alternative Flow #2: If at anytime the User selects an existing appointment, the following happens: 1. User selects existing appointment 2. System displays information on the existing appointment ● Information includes: date, time, location, doctor, hospital, and a short description 3. System gives options to edit and cancel ● the cancel option is only available for doctors and nurses 4. User closes detailed view 5. System returns to main calendar ● The main calendar is the week view
Post Condition:	Success: Appointments are reflected in Appointment Calendar

Use Case Number:	UC-09
Use Case Name:	Add/Remove Prescriptions
Overview:	Allows a doctor to add or remove prescriptions
Actor(s):	Doctor and admin
Pre-condition(s):	● System has been set up and accepts operation. ● User has login into the account ● User has permission to access to change the prescriptions
Scenario Flow:	1. User choose the patient 2. User opens the prescription page for a patient 3. System returns all prescriptions of the selected patient 4. User selects one or multi prescriptions 5. User selects operation for chosen prescriptions ● Operations include add operation and remove operation 6. System check if the operation is available ● If there is no prescriptions available for the patient, there will be an

	error message came out. 7. System requests prescriptions information 8. User input prescriptions information 9. System checks if the information is not empty ● If the information is empty, system will return an error message and require user to enter the information or cancel the operation. 10. User confirm 11. System add or remove the prescriptions 12. System send updates to the patient
Alternate Flows:	Alternate Flow#1. user select no prescriptions and choose to remove. start at step 5. 1. System check if there is any prescriptions chosen 2. System show error message to the user 3. Back to step 3 in scenario flow
Post Condition:	If any changes were made by the user, the patient will receive a message informing to check the changed prescription lists.

Use Case Number:	UC-10
Use Case Name:	Viewing Patient Medical Information, Prescriptions and Tests and Results
Overview:	Allows a doctor or patient to view medical information, prescriptions and tests and results
Actor(s):	Doctor, nurse, admin and patients
Pre-condition(s):	● system has been set up and accepts operation. ● user has login into the account ● user has permission to access to patient's information
Scenario Flow:	1. User login to the system 2. User choose the patient from the list 3. System check if the user have access to the patient information ● An error message will be returned if the information is not available to current user. 4. System returns all informations of the selected patient
Alternate Flows:	Alternate Flow#1. user cannot view the patient information. start at step 2. 1. System check if the user has permission to view the patient information

	● Patient can only view their own information. If the login account is a patient type, system will skip step 2 and directly return the information of the account owner. 2. System show error message to the user 3. Back to step 2 in scenario flow
Post Condition:	Users are able to view the information of the patient.

Use Case Number:	UC-11
Use Case Name:	Release Test Results
Overview:	Doctors (within the patient's hospital) can, upon evaluating a patient's test results, release them for view by that patient. Comments may be added to the specific test result for view by the patient.
Actor(s):	Doctor, Patient
Pre-condition(s):	● Doctor is logged into system as themselves. ● Patient has completed test(s). ● Test information is already uploaded to the system in the correct format. ● Both doctor and patient are registered in the same hospital.
Scenario Flow:	Main (success) Flow: 1. Doctor fetches patient test(s) from database. 2. Doctor tries to send patient test information to patient profile. 3. System verifies both doctor and patient are in the same hospital. 4. Patient profile is updated with patient's test information. 5. System notifies doctor that the test is successfully updated.
Alternate Flows:	Alternate Flows: ● Step 1 - System displays notification if it can't find any tests for the specified patient. ● Step 2 - System denies test release if patient belongs to a different hospital ● Step 4 - System fails to upload test if format is wrong. ○ Step 5 - System notifies doctor that test update is unsuccessful. ● Step 1 - Doctor can choose to add comments to the patient's test results prior to sending them.
Post Condition:	Patient is able to view test information on their own profile (must be logged in to their own account).

Use Case Number:	UC-12

Use Case Name:	Logging System Activity
Overview:	For security, many actions in the system will be logged for review at a later date, including, but not limited to, viewing/updating a Patient's information and transfers of a Patient between hospitals.
Actor(s):	Patient, Nurse, Doctor, Admin
Pre-condition(s):	● System is setup and configured, ready to log relevant activity ● All relevant activity is running correctly (viewing info, updating info, etc.)
Scenario Flow:	Main (success) Flow: 1. User completes activity 2. System determines activity is necessary to log for later review 3. Activity is logged in system database, viewable only by Admin ● Contains pertinent info (i.e. type, date, user, description, if applicable) 4. System returns to previous state (i.e. logging is completed in the background)
Alternate Flows:	Alternate Flow #1: If system determines activity is unnecessary to log in step 2 1. System determines activity unworthy of logging 2. No further action is taken ● Activity is complete, no logging has taken place 3. System returns to previous state (activity is complete)
Post Condition:	Success: Activity has been logged by the system to be reviewed at a later time. Otherwise: Activity has been deemed unnecessary of a logging, and no additional action is taken. Activity is not logged.

Use Case Number:	UC-13
Use Case Name:	Admission and Discharge to/from Hospital
Overview:	Doctors and nurses can admit patient to hospital for extended stay, event recorded by system. Only Doctors can approve of Patient discharge, recorded by system.
Actor(s):	Patient, Nurse, Doctor
Pre-condition(s):	● System is setup and configured, ready to log relevant activity ● All relevant activity is running correctly (viewing info, updating info, etc.)
Scenario Flow:	Main (success) Flow: 1. Patient arrives at Hospital 2. No appointment scheduled ● Emergency situation

	3. Doctor / Nurse examine Patient 4. Doctor / Nurse determine Patient must be admitted to Hospital 5. Doctor / Nurse logs into account 6. Doctor / Nurse admit Patient to Hospital ● Recorded by system 7. Doctor / Nurse logs out of account 8. Patient undergoes extended stay 9. Doctor assesses Patient 10. Doctor determines Patient is ready for discharge 11. Doctor logs into account 12. Doctor approves Patient discharge ● Event recorded by system 13. Patient is discharged 14. Patient Info is Updated ● Event recorded by system 15. Doctor logs out of account
Alternate Flows:	Alternate Flow #1: If patient has appointment in step 2 1. Patient shows up for appointment 2. Doctor / Nurse logs in to account 3. Doctor / Nurse view Appointment Calendar ● Verify Appointment 4. Patient attends Appointment 5. Doctor decides to have Patient undergo extended stay 6. Continue from step 5 of Main flow above Alternate Flow #2: If Doctor determines Patient is unfit for discharge in step #10: 1. Patient Information is Updated 2. Patient continues stay ● Until Doctor deems Patient fit for discharge 3. Continue from step #11 in Main flow above
Post Condition:	Success: Admission / discharge has been handled and recorded by system Otherwise: Activity has been deemed unnecessary of a logging, and no additional action is taken. Activity is not logged.

Use Case Number:	UC-14
Use Case Name:	Viewing Activity Log
Overview:	Administrators will be able to view the logs of all system activity for a given time-frame at their hospital
Actor(s):	● Administrator

Pre-condition(s):	● System is setup and configured ● System is operating and unmodified ● User has valid Admin access
Scenario Flow:	Main (success) Flow: 1. Administrator (Admin) opens system login page 2. System requests login credentials 3. Admin enters valid login credentials 4. Home page displays available options. 5. Admin selects "View recent system activity" ● Any hospital activity is logged by date in descending order from current date. 6. System requests a start and end date window for logged activity. 7. Admin selects a start and end date for time-frame of logged activity 8. System page displays all logged system activity 9. System presents multiple activity sort options 10. Admin views all relevant logged activity 11. Admin returns to system home page 12. Admin logs out.
Alternate Flows:	1. If at 3 the admin enters invalid credentials system denies access and requests the correct credentials again 2. After step 9 admin can: 1. sort by patient name in ascending order. 2. sort by date in descending order from end time 3. sort by date in ascending order from start time
Post Condition:	Success: Admin is able to view all relevant system activity.

Use Case Number:	UC-15
Use Case Name:	Viewing System Statistics
Overview:	Administrators will be able to view compiled statistics for a given time-frame at their hospital
Actor(s):	● Administrator
Pre-condition(s):	● System is setup and configured ● System is operating and unmodified ● User has valid Admin access
Scenario Flow:	Main (success) Flow: 1. Administrator (Admin) opens system login page 2. System requests login credentials 3. Admin enters valid login credentials

	4. System displays available options for Admin 5. Admin selects "View System Statistics" ● Any recorded data in the hospital is put on a database 6. System displays several statistics options including: ● Number of patients ● Average number of visits per patient ● Average patient's length of stay ● Most commonly listed reason for patient's visit ● Prescriptions listed 7. Admin chooses a relevant statistic. 8. System page displays all relevant statistics 9. Admin views statistics. 10. Admin returns to system home page 11. Admin logs out.
Alternate Flows:	1. At step 7 admin may choose to display multiple statistics at once ● they may also choose particular patient, doctor or nurse statistics to view
Post Condition:	Success: Admin is able to view all relevant system activity.

Use Case Number:	UC-16
Use Case Name:	Patient Transfer
Overview:	Patient can be transferred between hospitals. Transfers can be carried out by either administrators or by doctors (ones who are at the receiving hospital).
Actor(s):	● Administrator ● Doctor
Pre-condition(s):	● System is setup and configured ● System is operating and unmodified ● User is logged in as Administrator or Doctor
Scenario Flow:	1. User select the patient waiting to transfer 2. User select the Transfer Patient function 3. User input the new hospital waiting to transfer into a. If the hospital is not exist, user cannot continue to save unless he input the correct hospital. 4. User confer the Changes and save
Alternate Flows:	● In step 3 the system will check if the input is valid. If not, an error message will appear on the screen if the user trying to save the changes. Only when

	the input is valid the user can save the changes into the system.
Post Condition:	Success: The hospital information of the patient will change to updated one. Fail: The information of the patient will not change.

Use Case Number:	UC-17
Use Case Name:	Upload Patient Information
Overview:	Doctors will be able to upload the results of a patient's tests if needed, as well as images such as those used in X-Rays to update a patient's record.
Actor(s):	Doctor
Pre-condition(s):	● Patient exists and had the test(s) performed ● Doctor is logged in
Scenario Flow:	1. Doctor receives test results / relevant images 2. Doctor accesses Patient's profile 3. Doctor clicks UPDATE MEDICAL INFORMATION button 4. Doctor selects UPLOAD FILES button option 5. System opens up File Explorer system 6. Doctor selects relevant files and uploads them to profile 7. To finalize the updates, Doctor must hit SAVE button 8. Doctor re-enters login information to confirm changes 9. System saves changes to medical information
Alternate Flows:	Alternate Flow #1: If at step #2, patient is inactive: 1. System displays error message and disallows updating the medical information
Post Condition:	On system success the patient's medical information is updated, with the added files not visible by patient until review period is up and files are confirmed to be correct

Use Case Number:	UC-18
Use Case Name:	Private Messaging System
Overview:	Doctors, nurses, patients and administrators can send private messages of limited length via the system.
Actor(s):	● Doctor ● Nurses ● Patients ● Administrators

Pre-condition(s):	● User is logged and account is active ● Recipient account is valid ● User can navigate Healthnet
Scenario Flow:	Main (success) Flow: 1. User goes to Recipient Profile and hits send message 2. System request message 3. User input message 4. User hits send 5. System checks if message is valid and of correct length ● If message isn't valid, System requests a new message. 6. System sends message to Recipient
Alternate Flows:	Alternate Flow #1: This flow is for a user receiving a message: 1. Users selects view messages button 2. System gets messages 3. System displays current conversations 4. User selects conversations 5. System loads conversation with newest message at top 6. User hits reply 7. System continues to main flow at step 2 Alternative Flow #2: This flow occurs if user cancels sending a message at any time 1. System confirms intend to cancel 2. User confirms intend 3. System returns to messages page.
Post Condition:	On success system sends message and returns to messages page On fail system doesn't send message and returns to messages page

Use Case Number:	UC-19
Use Case Name:	Patient Prescription Refills
Overview:	Authenticates Patient prescription and insurance information. If pharmacy has the drug shows the price and pickup time.
Actor(s):	● Patient
Pre-condition(s):	● Patient has a prescription from a Doctor. ● Patient has insurance to cover the prescription. ● Patient is logged in as a Patient.
Scenario Flow:	Main (success) Flow:

	1. Patient navigates to the list of their prescriptions. 2. Patient clicks on the prescription that they want to refill. 3. Patient clicks the refill button. 4. System checks the patient's prescription. 5. System checks the patient's insurance. 6. The System notifies Pharmacy that the Patient is refill their prescription. 7. The Pharmacy returns the price of the prescription and the time that the prescription will be ready to the System. 8. The System notifies the Patient of the price and the pickup time. 9. Patient picks up the the prescription at or after the pickup time and pays for the prescription at the Pharmacy.
Alternate Flows:	Alternate Flow #1: After step 4 if the prescription expired or refill maximum is reached. 1. The System notifies the Patient and ends the refill process. Alternate Flow #2: After step 5 if the insurance does not cover the prescription. 1. The System notifies the Patient and ends the refill process.
Post Condition:	The Patient's prescription is refilled.