**Requirements Document**

**Date:** May 6, 2015

**Version: 1.12**

**Team: Team - A**

**1.0 Brief Problem Statement**

Hospitals across the country lack a comprehensive, easy-to-use technological method of communication between the hospital, its employees, and the patients. Our solution to this problem is to create a new web application called **HealthNet**. HealthNet will facilitate the communication necessary to improve health care for its users while still allowing staff to manage permissible information more efficiently than currently-available systems.

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.

**2.0 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.

**3.0 Users profile**

**Computer System Administrators:** Users that will have administrative abilities over the software. They will have access to non-private user account information, such as user names, and technical details, as well as anonymous metrics and statistical information. System Administrators will be capable of designing information fields for all accounts.

**Business Administrators:** Are users that will have the ability to access and manage the statistical data and oversee the usage of the HealthNet System.

**Secretaries:** Are a member of the Business staff. They have access to patient non-medical information, all calendars, billing information, and appointments at their hospitals.

**Doctors:** A user that will have access to the medical information of all of their patients as well as the ability to prescribe pharmaceuticals. Doctors will be able to share patient information with other doctors (surgeons, specialists). Doctors are also the only account that can release a patient’s test data for their viewing.

**Medical Staff accounts:** Accounts that will have access to all patient’s drug allergies, as well as access to patient’s medical information as permitted by the doctor for which they serve. These account are primarily for the nurses that work at each hospital.

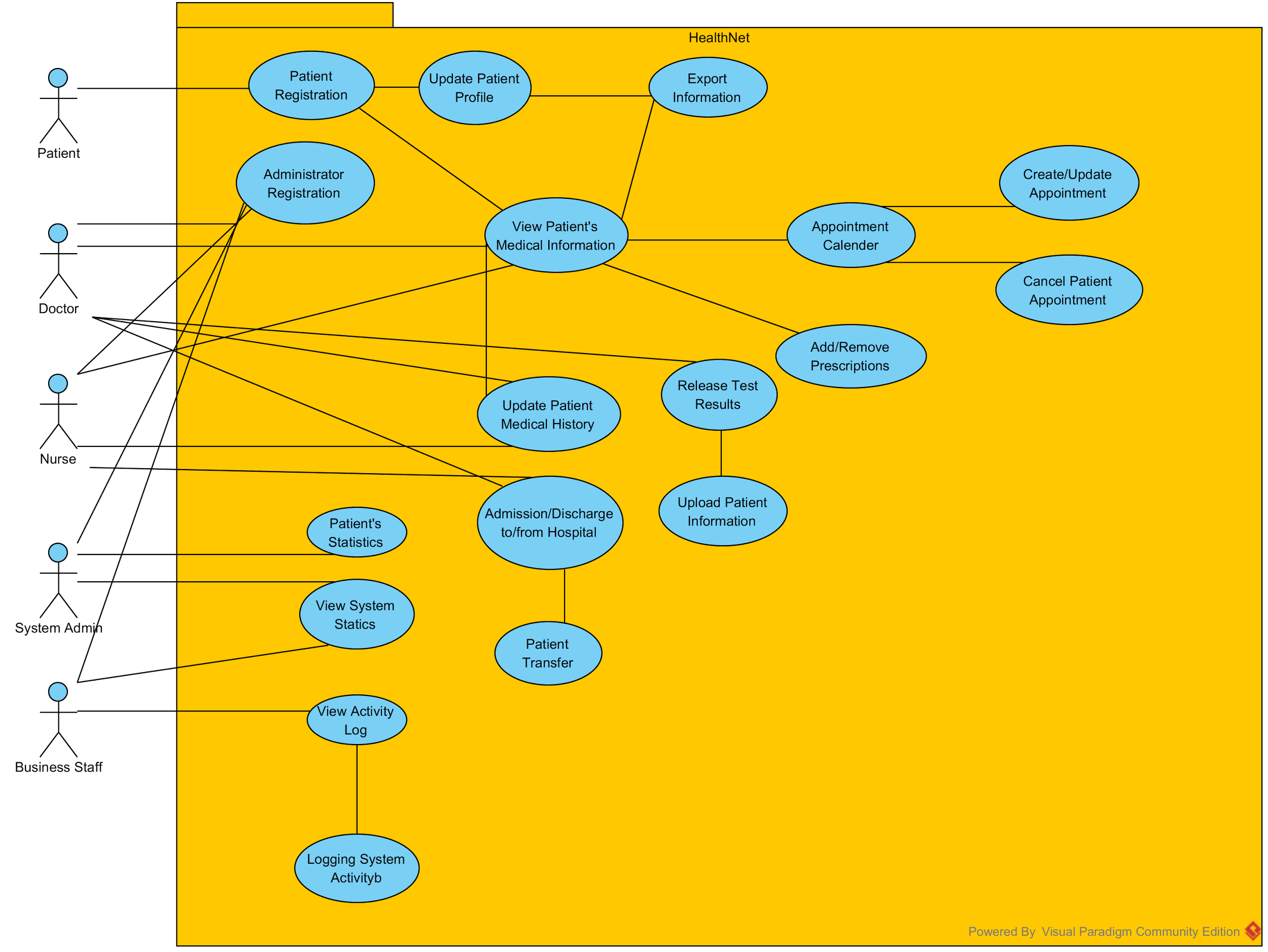
**Patient:** Basic account that will allow users to enter simple information about themselves such as medical history, allergies, and other relevant information. They can view their test results after they are released by the Doctor. They can also view any appointment dates that they may have scheduled.

**4.0 Use Cases (Operational Scenarios)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Number** | **User Story Name** | **Description** | **Release** |
| UC-01 | Patient Registration | Users sign up to become a Patient with their personal contact information and creates login credentials.  Additionally, a patient may provide the system with some basic medical profile information, a choice of preferred hospital, health insurance provider, and emergency contact information (linked to another patient if they are already in the system). | R1 |
| UC-02 | 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 the administrator console. | R1 |
| UC-03 | Update Patient Profile Information | Patients can update their profile information. | R2 |
| UC-04 | Update Patient Medical Information | Doctors and Nurses can update patient medical information. | R2 |
| UC-05 | Export Information | Patients will be able to export their information and their test results from the system with relevant privacy warnings.  Information will be available in csv format. | X |
| UC-06 | 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.  If the patient or doctor already has an appointment at the time selected, then the system will not allow for the appointment. | R2 |
| UC-07 | Cancel Patient Appointment | Patients can cancel their existing appointments.  Doctors can cancel their existing appointments.  Nurses cannot cancel (only modify)  existing appointments. | R2 |
| UC-08 | 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. | R2 |
| UC-09 | 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. | R1 |
| UC-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. | R2 |
| UC-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.  Comments may be added to the specific test result for view by the patient. | R2 |
| UC-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 Patients information/records, and transfers of a Patient from one hospital to another. | R2 |
| UC-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 |
| UC-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. | R2 |
| UC-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 hospital * prescription statistics * anonymous patient ages * anonymous patient genders * anonymous patient visit locations * hours worked by each doctor by name * hours worked by nurses and staff by name * hours each facility is open   Other important and informative statistics yet to be determined. | R2 |
| UC-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 |
| UC-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 |
| UC-18 | Send Private Message | Doctors, nurses, patients and administrators can send private messages of limited length via the system. | R2 |
| UC-19 | Patient’s Statistics | Doctors can log what each patient's personal health stats are | R2 |
| UC-20 | Password Recovery | A user can specify recovery questions for their account.  If the user forgets his or her password, he or she can navigate to a password recovery page, enter their username and email, and respond to the questions. If the responses are the same as what the user has specified, the user will be prompted to create a new password. | X |

**4.1 Use case diagram**

The follow diagram shows the name of each Use Case Name and Actors that have a Major Role in that Use Case’s activity. The Use Cases are better described in 4.2 of this Document.



**4.2 Use Cases (Description)**

|  |  |
| --- | --- |
| **Use Case Number:** | **UC-01-01** |
| **Use Case Name:** | Patient Registration |
| **Overview:** | Registrant shall provide personal to the System upon registering and becoming a Patient. |
| **Actor(s):** | Registrant/ Patient |
| **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. 5. System displays confirmation of registration |
| **Alternate Flows:** | Main (success) Flow:  **Alternate #1:** The information that the Registrant provides is incorrect.   1. The system returns the user to re-enter there personal information and also provides them with information that was incorrectly entered 2. Registrant re-enters all the information 3. System verifies required information is provide 4. System displays confirmation of registration |
| **Post Condition:** | If registrant complete registration, system stores Registrant's information. |

|  |  |
| --- | --- |
| **Use Case Number:** | **UC-02-01** |
| **Use Case Name:** | Administrator Registration |
| **Overview:** | A System Admin will add registrant to the system, specifying only log-in information and the registrant’s e-mail address to send such information.  Registrant will log in with this information and proceed to change its password. Registrant will then provide personal, medical, and emergency contact information to the System upon registering and becoming an Administrator, a Nurse, or a Doctor. |
| **Actor(s):** | Registrant/Administrator, Doctor, or Nurse  Registrar/Administrator |
| **Pre condition(s):** | - There is at least one registered Administrator in the system  - System has been setup and configured  - System is running and open for registrations |
| **Scenario Flow:** | Main (success) Flow:   1. Registrar logs into the system 2. Registrar navigates interface to ‘Add an Administrator’ 3. System requests a new user e-mail and relevant information 4. Registrar enters Registrant’s e-mail and relevant information 5. System verifies if e-mail is already in the system    * If user e-mail already exists in the system, repeat Step 3 6. System prompts a confirmation box to confirm the intended registrant is to be added    * If user selects OK, go to step 7    * If users selects cancel, go to step 3 7. Registrar enters and confirms an initial password for the user. 8. Registrar notifies registrant of account creation 9. Registrant logs in using registrar-provided authentication details 10. System recognizes Registrant’s first log in 11. System requests new password 12. Registrant provides new password. 13. System changes registrant password 14. System displays confirmation of registration |
| ***Alternate Flows:*** | Main (success) Flow:  **Alternate #1**: After Step 3, System will display the option to Cancel the registration process:   1. Registrar selects option to cancel during registration 2. System requests confirmation to cancel 3. Registrar confirms intent 4. System returns to home screen   Post Condition: Registrar did not complete registration. System does not store Registrant’s information. |
| **Post Condition:** | Registrar completes registration. System stores Registrant's information. |

|  |  |
| --- | --- |
| **Use Case Number:** | **UC-03-01** |
| **Use Case Name:** | Update Patient Profile Information |
| **Overview:** | Patient will update their profile information with current medical information. |
| **Actor(s):** | Patient |
| **Pre condition(s):** | - System has been setup and configured.  - System is running and open for registrations.  - Patient has already registered  - Patient is logged into their account. |
| **Scenario Flow:** | ***Main (success) Flow:***   1. System requests login 2. System verifies if password is correct  * If password is invalid System displays wrong password and prompts the user to try again  1. System shows current medical information with the ability to edit each entry 2. User navigates to profile page 3. User selects edit data 4. Patient updates necessary medical information 5. System verifies required information is provided and correct.  * If information is invalid System displays message and returns to Step 5.  1. System verifies if password is correct  * If password is invalid System displays wrong password and prompts the user to try again  1. System displays confirmation of edit |
| ***Alternate Flows:*** | **Main (success) Flow:**  **Alternate #1:**After Step 5 in success scenario System will display the option to Cancel the edit process. The following steps would occur:   1. Patient selects option to cancel during the editing of the information 2. System prompts the Patient to confirms cancellation 3. Patient confirms cancel 4. System returns to information screen |
| **Post Condition:** | Patient did not complete edits. System does not store new updated information. |

|  |  |
| --- | --- |
| **Use Case Number:** | **UC-04-01** |
| **Use Case Name:** | Update Patient Profile Information |
| **Overview:** | *Doctors and nurses will be able to update their patients’ medical information with current medical information.* |
| **Actor(s):** | *Doctors and Nurses* |
| **Pre condition(s):** | - System has been setup and configured.  - System is running and open for registrations.  - Patient has already registered  - User is a patient of editing Doctor or Nurse  - Editor is logged in |
| **Scenario Flow:** | **Main (success) Flow:**   1. Editor navigates to patient profile 2. Editor selects edit 3. System prompts Editor with log-in 4. Editor provides log-in details 5. System verifies log-in information  * If password is invalid System displays wrong password and prompts the user to try again  1. System shows the patient’s current medical information with the ability to edit each entry 2. Editor updates patient’s necessary medical information 3. System verifies required information is provided and correct.  * If information is invalid System displays message and returns to Step 5.  1. System checks log-in again  * If password is invalid System displays wrong password and prompts the user to try again  1. System displays confirmation of registration 2. System returns to patient’s updated information page |
| ***Alternate Flows:*** | **Main (success) Flow:**  **Alternate #1:** After Step 5 in success scenario System will display the option to Cancel the edit process. The following steps would occur:   1. Editor selects option to cancel during the editing of the information 2. System prompts the Editor to confirms cancellation 3. Editor confirms cancel 4. System returns to information screen |
| **Post Condition:** | Editor did not complete edits. System does not store new updated information. |

|  |  |
| --- | --- |
| **Use Case Number:** | **UC-05-01** |
| **Use Case Name:** | Export Information |
| **Overview:** | Patients will be able to export their information and their test results from the system with relevant privacy warnings. |
| **Actor(s):** | Patient |
| **Pre-**  **conditions(s)** | * System has been setup and configured * Patient has accessed website via URL * Patient is registered with the site * Patient has seen their Doctor and has information to export |
| **Scenario**  **Flow:** | 1. Patient goes to the site 2. Clicks to log-in 3. Patient enters their log-in information 4. The system accepts their log-in information 5. Patient navigates to view their information. 6. Patient clicks “Export” 7. Patient is prompted by the system about the privacy risks 8. Patient selects “Accept” 9. Then the Patent's information is packaged and downloaded to the user’s local hard drive for their viewing and usage |
| **Alternate**  **Flows:** | **Alternate #1:** At step 3, when the Patient enters their log-in information   1. The Patient enters wrong/invalid log-in information 2. System prompts the Patient that the log-in information is in-correct and also displays a “Click here to Register” Link to take them to the Registration Page.   **Alternate # 2:** At step 8, the Patient clicks “Decline” on the privacy prompted   1. The Patient is returned to the previous window |
| **Post**  **Conditions:** | Patient now has a local copy of his/her medical history. |

|  |  |
| --- | --- |
| **Use Case Number:** | **UC-06-01** |
| **Use Case Name:** | Create or Update Patient Appointment |
| **Overview:** | Patient can make an appointment with a doctor for medical check-up (or) specialty care. |
| **Actor(s):** | Patient, Nurse, Secretaries and Doctor |
| **Pre condition(s):** | - System has been setup and updated.  - System is running and open for scheduling an appointment.  - Registrant should bring health insurance card (or) information |
| **Scenario Flow:** | **Main (success) Flow:**   1. User Logs in 2. User clicks profile 3. User clicks “Appointment” 4. User selects “New Appointment” 5. User provides the form with the following information: User, Doctor, Name, Date, Time and Description 6. System validates and adds the appointment to the Patients Calender |
| ***Alternate Flows:*** | **Main (updating an appointment) Flow:**  **Alternate #1:** Information does not validate successfully.   1. System takes the user back to the create appointment page to re-enter the information 2. User re-enter information 3. User submits data 4. Data is re-validated by the System 5. System updates the patients appointments |
| **Post Condition:** | If registrant complete scheduling, system stores Registrant's information and send notification to doctor’s calendar. |

|  |  |
| --- | --- |
| **Use Case Number:** | **UC-07-01** |
| **Use Case Name:** | Cancel Patient Appointment |
| **Overview:** | Patient cancels the appointment with a doctor that has been in the system. |
| **Actor(s):** | Patient, Doctor, Nurse |
| **Pre condition(s):** | - System has been setup and updated.  - System is running and open for scheduling an appointment.  - An existing appointment has been scheduled. |
| **Scenario Flow:** | **Main (success) Flow**:   1. User logs in and selects profile 2. User Clicks on “Appointment” 3. User uses the calender to navigate to there appointment and clicks the date 4. User then clicks the Appointment 5. User is prompted with options to modify, delete or go back 6. User Clicks delete 7. System updates the calender and the patients appointments to reflect the change. |
| **Post Condition:** | If registrant complete cancelling appointment, system delete the appointment on Registrant's information and send notification to doctor’s calendar. |

|  |  |
| --- | --- |
| **Use Case Number:** | **UC-08-01** |
| **Use Case Name:** | Appointment Calendar |
| **Overview:** | The Calendar displays all of the appointments in a calendar view. All employees in a hospital will be able to see all appointments for the day and week between Patients and Doctors. |
| **Actor(s):** | Nurses, Doctor, and Patient |
| **Pre condition(s):** | - The calendar has been setup and updated.  - The calendar is running and open for scheduling an appointment.  - The calendar is able to be edited by doctors/ nurses/ system administrators. |
| **Scenario Flow:** | Main Description:   1. User logs in and navigates to the Profile Page 2. User click “Appointment” 3. System takes the user to the calender. 4. Calender opens to the current month and will show any appointments that the user may have that month 5. The user can use the arrows to go to the next month or a previous one 6. User clicks the date that there is an appointment and then selects the desired appointment to view the details of the appointment |
| **Post Condition:** | The time-management calendar displays the appointments and events in a weekly or monthly basis where system administrator and nurses can create and cancel an appointment. |

|  |  |
| --- | --- |
| **Use Case Number:** | **UC-09-01** |
| **Use Case Name:** | Add Prescriptions |
| **Overview:** | Doctor will add a prescription into a patient’s account. |
| **Actor(s):** | Doctor |
| **Pre condition(s):** | - System has been setup and configured.  - Patient is registered in the system.  - Doctor is registered in the system. |
| **Scenario Flow:** | Main (success) Flow:   1. Doctor logs into account 2. Doctor looks up patient through one of these options:    1. Doctor navigates to ‘My Patients’ and locates the patient    2. Doctor uses search function to look up patient 3. On Patients profile, Doctor clicks on ‘Prescriptions’ 4. On ‘Prescriptions’, Doctor clicks on “Add Prescription” 5. System requests prescription information 6. Doctor enters prescription information and clicks ‘Save’ 7. System verifies required fields are filled    1. If required fields are filled, move on to Step 8    2. If required fields are missing, go back to step 5 8. System saves prescription in database 9. A confirmation box is shown |
| ***Alternate Flows:*** | Main (success) Flow:  **Alternate #1**: After Step 6, System will display the option to Cancel the add prescription process:   * Doctors selects option to cancel prescription * System returns to patient’s ‘Prescriptions’ page   Post Condition: Doctor did not add a prescription to the patient. |
| **Post Condition:** | Doctor adds a prescription to the patient. |

|  |  |
| --- | --- |
| **Use Case Number:** | **UC-09-02** |
| **Use Case Name:** | Remove Prescriptions |
| **Overview:** | Doctor will remove a prescription from a patient’s account. |
| **Actor(s):** | Doctor |
| **Pre condition(s):** | - System has been setup and configured.  - Patient is registered in the system.  - Doctor is registered in the system.  - There’s at least one prescription in Patient’s account. |
| **Scenario Flow:** | Main (success) Flow:   1. Doctor logs into account 2. Doctor looks up patient through one of these options:    1. Doctor navigates to ‘My Patients’ and locates the patient    2. Doctor uses search function to look up patient 3. On Patients profile, Doctors clicks on ‘Prescriptions’ 4. Doctor clicks the trash bin icon next to the prescriptions to be deleted 5. System asks for confirmation 6. Doctors clicks “Yes” 7. System deletes prescription from database 8. A confirmation box is shown confirming the prescription has been deleted |
| ***Alternate Flows:*** | Main (success) Flow:  **Alternate #1**: After Step 5, the confirmation box will contain the option to cancel the removal.   1. Doctors selects “Cancel” 2. System returns to patient’s “Prescriptions” profile. |
| **Post Condition:** | Doctor removes the desired prescription. |

|  |  |
| --- | --- |
| **Use Case Number:** | **UC-09-03** |
| **Use Case Name:** | View Prescriptions |
| **Overview:** | Doctor or Nurse views a Patient’s prescription. |
| **Actor(s):** | Doctor  Nurse |
| **Pre condition(s):** | - System has been setup and configured.  - Patient is registered in the system.  - Doctor or Nurse is registered in the system.  - There’s at least one prescription in Patient’s account. |
| **Scenario Flow:** | Main (success) Flow:   1. Doctor/Nurse logs into account 2. Doctor/Nurse looks up patient through one of these options:    1. Doctor/Nurse navigates to ‘My Patients’ and locates the patient    2. Doctor/Nurse uses search function to look up patient 3. On Patients profile, Doctor/Nurse clicks on ‘Prescriptions’ 4. System loads available prescriptions according to permissions 5. Doctor/Nurse clicks the magnifying glass icon next to the prescription that needs to be opened 6. System loads and displays the prescription |
| ***Alternate Flows:*** | --- |
| **Post Condition:** | Doctor/Nurse views the desired prescription. |

|  |  |
| --- | --- |
| **Use Case Number:** | **UC-09-04** |
| **Use Case Name:** | View Prescriptions |
| **Overview:** | A Patient views his prescriptions |
| **Actor(s):** | Patient |
| **Pre condition(s):** | - System has been setup and configured.  - Patient is registered in the system.  - There’s at least one prescription in Patient’s account. |
| **Scenario Flow:** | Main (success) Flow:   1. Patient logs into account. 2. On his profile page, the patient clicks on ‘Prescriptions’ 3. System loads available prescriptions according to permissions. 4. Patient is shown a list of prescriptions that are prescribed to him and can click on more for more detail. |
| ***Alternate Flows:*** | --- |
| **Post Condition:** | Patient views the desired prescription. |

|  |  |
| --- | --- |
| **Use Case Number:** | **UC-10-01** |
| **Use Case**  **Name:** | Viewing Patent's Medical Information, Prescriptions and Test Results. |
| **Overview:** | 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. |
| **Actor(s):** | Patient |
| **Pre-**  **Condition(s):** | * System has been setup and configured * System is running and able to log users in * User has access to the website via URL |
| **Scenario**  **Flow:** | 1. Patient selects option to view their information 2. System pulls logs of their prescriptions without any further approval 3. Patient clicks to see their test results 4. System pulls the data for that Patient from the database 5. Patient can see the statuses of their tests, “Pending” or “Released” 6. Patient Clicks “Released” 7. Patient is taken to a details page to display the test results |
| **Alternate**  **Flow:** | **---** |
| **Post**  **Condition(s):** | Patient has found the information he/she is entitled to view. |

|  |  |
| --- | --- |
| **Use Case Number:** | **UC-10-02** |
| **Use Case**  **Name:** | Viewing Patent's Medical Information, Prescriptions and Tests and Results. |
| **Overview:** | Doctors can view all medical information for any patient in the system (regardless of Hospital). |
| **Actor(s):** | Doctor |
| **Pre-**  **Condition(s):** | * System has been setup and configured * System is running and able to log users in * User has access to the website via URL |
| **Scenario**  **Flow:** | 1. Doctor clicks to view information on a Patient and is prompted by the system to enter search information about the Patient, which is open to ALL Hospitals 2. Doctor can see all information:    1. Test Results    2. Health History    3. Personal Contact Information    4. Prescriptions    5. Enter/Release Test Results |
| **Alternate**  **Flow:** | **Alternate #1:** The Doctor searches for a Patient that does not exist in the Database   1. The user is taken back to the search window 2. System displays that the user does not exist or has not yet register for the site |
| **Post**  **Condition(s):** | Doctor has found the information that he/she is entitled to view |

|  |  |
| --- | --- |
| **Use Case Number:** | **UC-10-03** |
| **Use Case**  **Name:** | Viewing Patent's Medical Information, Prescriptions and Tests and Results. |
| **Overview:** | Nurses can only view patient medical information in the hospital they work for. |
| **Actor(s):** | Doctor |
| **Pre-**  **Condition(s):** | * System has been setup and configured * System is running and able to log users in * User has access to the website via URL |
| **Scenario**  **Flow:** | 1. Nurse selects option to view information on a Patient 2. The system prompts the nurse to enter in search information about the Patient 3. The system looks for a Patient that matches the information that is limited to ONLY the Hospital that the nurse works at. |
| **Alternate**  **Flow:** | **Alternate #1:** The Nurse searches for a Patient that does not exist in the Database   1. The user is taken back to the search window 2. System displays that the user does not exist or has not yet register for the site   **Alternate #2:** The Nurse searches for a Patient that is not in their Hospital   1. The Nurse is taken back to the search window 2. System displays that “Patient *name* is not in your Hospital” |
| **Post**  **Condition(s):** | Nurse has found the information that he/she is entitled to view |

|  |  |
| --- | --- |
| **Use Case Number:** | **UC-11-01** |
| **Use Case Name:** | Release Test Results |
| **Overview:** | Doctors in the patient’s hospital can, upon evaluating a patient’s test results, release them for view by that patient. |
| **Actor(s):** | Doctor, Patient |
| **Pre condition(s):** | * System has been set up and configured * System is operational * Patient is registered to HealthNet * Doctor is assigned to Patient * Doctor has ordered a test for Patient * Patient has undergone said test * Doctor has access to results from test |
| **Scenario Flow:** | ***Main (success) Flow:***   1. Doctor logs in 2. Doctor navigates to page for patient test results 3. Doctor selects option to release results to patient 4. System displays confirmation dialogue 5. Doctor confirms release of test results 6. System provides information to relevant patient account 7. System notifies patient that test results are available |
| **Alternate Flows** | **Alternate #1**: After Step 3 in the success scenario, the system will provide the option for the doctor to append a comment to the test results. The following steps will occur:   1. Doctor selects option to comment on results 2. System provides a text field for comment 3. Doctor enters a comment into the text field 4. Doctor submits comment to system  * If the comment is too large for the system to accept, the system displays a message. Return to Step 2.  1. System appends a doctor’s comment field to information 2. System fills doctor’s comment field with provided comment 3. Process returns to Step 2 of success scenario and resumes   New Post Condition: Patient has access to test results. A comment from the doctor who provided information to the patient is also included.  **Alternate #2**: After Step 4 in the success scenario, the system will display an option to cancel the release process. The following steps will occur:   1. Doctor selects option to cancel test release 2. System request confirmation to cancel 3. Doctor confirms intent 4. System returns to original screen   New Post Condition: Patient does not have access to the test results. |
| **Post Condition:** | Patient has access to test results. |

|  |  |
| --- | --- |
| **Use Case Number:** | **UC-12-01** |
| **Use Case Name:** | Logging System Activity |
| **Overview:** | For security, many actions in the system will be logged for review at a later date. |
| **Actor(s):** | System Administrator |
| **Pre condition(s):** | * System has been set up and configured * System is operational * Actions are occurring within the system |
| **Scenario Flow:** | ***Main (success) Flow:***   1. System logs activity within the system as it occurs 2. System provides opportunity to system administrator to review log 3. System administrator is able to review log and flag possible security alerts |
| **Alternate Flows** | ***Alternate Flows:***  **Alternate #1**: The system may be interrupted while monitoring system activities. The following steps will occur:   1. System notifies system administrator of interruption 2. System administrator inspects system and solves source of interruption 3. Nature and duration of interruption is calculated by system after restoration of functionality 4. Administrator manually logs interruption into system with comment identifying reason for interruption   *New Post Condition:* System administrator is unable to monitor the system for the duration of the interruption. However, system log indicates interruption and system administrator can inspect source and time of interruption. |
| **Post Condition:** | System administrator is able to monitor the system and activities within the system for the purposes of information integrity and security. |

|  |  |
| --- | --- |
| **Use Case Number:** | **UC-13-01** |
| **Use Case Name:** | Admission to Hospital |
| **Overview:** | Doctors and Nurses can admit a patient to the hospital for an extended stay. |
| **Actor(s):** | Doctors and Nurses |
| **Pre condition(s):** | - System has been setup and configured.  - System is running and open for registrations.  - Patient has already registered  - Employee is logged in |
| **Scenario Flow:** | ***Main (success) Flow:***   1. Employee navigates to Hospital Admission/Discharge page 2. Employee enters in an admission form with patients name 3. System admits the patient into the hospital and logs the information |
| **Post Condition:** | Patient is admitted to the hospital |

|  |  |
| --- | --- |
| **Use Case Number:** | **UC-13-02** |
| **Use Case Name:** | Discharge from Hospital |
| **Overview:** | Doctors are the only ones to approve a patient’s discharge from the Hospital. |
| **Actor(s):** | Doctors |
| **Pre condition(s):** | - System has been setup and configured.  - System is running and open for registrations.  - Patient has already registered and been admitted  - Doctor is logged in |
| **Scenario Flow:** | **Main (success) Flow:**   1. Doctor navigates to Hospital Admission/Discharge page 2. Doctor determines what patient is ready to be discharged 3. Doctor search name of Patient 4. System updates the record showing if the patient can be discharged and logs the information |
| **Post Condition:** | Patient is discharged from the hospital |

|  |  |
| --- | --- |
| **Use Case Number:** | **UC-14-01** |
| **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. 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. |
| **Actor(s):** | System Administrator, Business Administrator |
| **Pre condition(s):** | * System has been set up and configured * System is operational * Administrator has access to the system * System activity has been successfully logged |
| **Scenario Flow:** | ***Main (success) Flow:***   1. Administrator logs in 2. System displays option at administrator index page to show logs 3. Administrator selects option to show logs 4. System display separates types of logs into the following categories:    1. Viewing activity of patient records, sorted by date    2. General system activities, sorted by popularity 5. Administrator selects a log for system activity 6. System displays a full log corresponding to administrator’s selection |
| **Alternate Flows** | **Alternate Flows:**  **Alternate #1**: After Step 4 in the success scenario, the system displays an option to change how logs are ordered for each category. The following steps will occur:   1. Administrator selects drop-down box to change order for a category 2. System displays drop-down box with options for each value stored in a log 3. Administrator selects an option from the drop-down box   *New Post Condition*: Administrator can view information within system logs pertaining to their account type. Logs in each category are ordered as specified by administrator.  **Alternate #2**: After Step 4 in the success scenario, the system displays a Search text box for logs. The following steps will occur:   1. Administrator selects search box 2. Administrator types in a search query 3. Administrator confirms search query 4. System displays only logs that match the search query.   *New Post Condition*: Administrator can view information within system logs pertaining to their account type. Logs displayed to administrator are filtered based on search query. |
| **Post Condition:** | Administrators can view information within system logs pertaining to their account type. |

|  |  |
| --- | --- |
| **Use Case Number:** | **UC-15-01** |
| **Use Case Name:** | Viewing System Statistics |
| **Overview:** | Administrator will view a compiled statistical data for a time frame concerning their hospital. This includes, but is not limited to, number of patients visiting the hospital, average number visits per patient, average length of stay, reasons of stay, prescription statistics, etc.  More important information to be determined. |
| **Actor(s):** | Administrator |
| **Pre condition(s):** | - System has been setup, configured and is running.  - System has logged all relevant information required to produce statistical data. |
| **Scenario Flow:** | Main (success) Flow:   1. Administrator logs into the system. 2. Administrator navigates to “Statistics” 3. System renders the interface for statistics. 4. System prompts Administrator to specify a *Start Date* and an *End Date*. 5. Administrator specifies both dates and clicks ‘OK’.    1. If *End Date* occurs before *Start Date*, display error message and go back to step 4 6. System renders statistical data for said time frame. |
| ***Alternate Flows:*** | - - - |
| **Post Condition:** | Relevant statistical data for the given time frame is produced in a clean interface. |

|  |  |
| --- | --- |
| **Use Case Number:** | **UC-16-01** |
| **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):** | Doctor, Administrator |
| **Pre**  **Condition(s):** | * System has been setup and configured * System is running and open for logging in * User has accessed website via URL * New host hospital has been selected |
| **Scenario**  **Flow:** | 1. User goes to the site via URL 2. User clicks to log-in 3. System asks the user for log-in information 4. User enters the required log-in information 5. User selects a Hospital 6. User clicks on Transfer Patient 7. User selects the new Hospital and the patient to transfer 8. System updates the Patient list of both hospitals |
| **Alternate Flows:** | **Alternate #1:** User enter invalid log-in information   1. User has enter incorrect log-in information 2. System takes the user back to the log-in screen 3. System displays that the log-in failed and displays a link “Click here to Register” that will take the user to the Register page if click   **Alternate #2:** User enter incorrect/uncompleted transfer details   1. System takes the user back to the transfer page 2. System displays error message to the user 3. User corrects and tries again   **Alternate #3:** System fails to update the Database   1. User is taken back to the “User Panel” 2. System displays error message to the user that the transfer was not completed successfully and needs to try again later with and error message 3. User should record error message and notify the System Administrator of the error |
| **Post Condition(s):** | User successfully transfers a Patient into their hospital |

|  |  |
| --- | --- |
| **Use Case Number:** | **UC-17-01** |
| **Use Case Name:** | Upload Patient Information |
| **Overview:** | Doctors will be able to upload the results of a patient’s tests |
| **Actor(s):** | Doctor |
| **Pre condition(s):** | - System has been setup, configured and is running.  - Doctor is registered in the system  - Patient is registered in the system  - Doctor is logged in |
| **Scenario Flow:** | Main (success) Flow:   1. Doctor navigates to patients information page and selects upload test or image 2. System opens a window to browse for test/image 3. User selects file 4. System prompts for log-in  * If password is invalid System displays wrong password and prompts the user to try again  1. File is uploaded to the system 2. Doctor can then check if the patient is able to see the file or not. |
| **Post Condition:** | File is correctly uploaded to the patient’s medical information. |

|  |  |
| --- | --- |
| **Use Case Number:** | **UC-18-01** |
| **Use Case Name:** | Send Private Message |
| **Overview:** | Actor sends a private, limited-length message to another user. |
| **Actor(s):** | Administrator, Doctor, Nurse, and Patient |
| **Pre condition(s):** | - System has been setup, configured and is running.  - Actor is registered in the system  - Receptive user is registered in the system. |
| **Scenario Flow:** | Main (success) Flow:   1. Actors logs into the system. 2. Actor navigates to ‘My Contacts’ 3. System retrieves Actor’s contacts 4. Actor clicks on Envelope icon next to desired user. 5. System loads the *Compose Message* interface, with the *To*: field already directed for the desired user. 6. System prompts Actor for subject and message’s contents. 7. Actor enters subject and contents, then clicks “Send” 8. Message is sent to the user. |
| ***Alternate Flows:*** | Main (success) Flow:  **Alternate #1:** After step 1, Actor uses *Search* function to look for the receptive user.   1. Actor uses *Search* function to find receptive user. 2. On their profile, Actor clicks on *Send Message.* 3. Proceed to step 5 of the Main (success) flow.   **Alternate #2:** After step 1, Actor goes to mailbox to reply an existing mail conversation.   1. Actor goes to *Mailbox.* 2. System loads the mailbox interface. 3. Actor opens desired message. 4. Actor clicks on *Reply.* 5. Proceed to step 5 of the Main (success) flow.   **Alternate #3:** After step 1, Actor goes to mailbox to compose a new message.   1. Actor goes to *Mailbox*, 2. System loads the mailbox interface. 3. Actor clicks in *New Message*. 4. System loads the *Compose Message* interface 5. System prompts Actor for receptor’s user name, a subject, and message’s contents 6. Actor enters receptor’s user name, a subject, and contents. Actor clicks “Send”. 7. System checks if username exists.    * If username exists, proceed to step 8.    * If username doesn’t exist, clear the receptor’s user name field and prompt for it again. Then go back to step 6. 8. Message is sent to the user. |
| **Post Condition:** | Message is successfully sent to the intended user. |

|  |  |
| --- | --- |
| **Use Case Number:** | **UC-19-01** |
| **Use Case Name:** | Patent's Statistics |
| **Overview:** | Doctors can log what each patent's personal health states are |
| **Actor(s):** | Doctor |
| **Pre Condition(s):** | * System has been setup and configured * System is running and ready for user log in * User has accessed website via URL |
| **Scenario Flow:** | 1. Doctor goes to the site and clicks log in 2. System directs Doctor to the log in form 3. Doctor enters required log in inform 4. System take Doctor to “User Panel” 5. Doctor searches for a Patient 6. System returns the found Patient profile 7. Doctor clicks Patient States 8. System builds and returns a form with all of the patients states are |
| **Alternate Flows:** | **Alternate #1:** User enter invalid log-in information   1. User has enter incorrect log-in information 2. System takes the user back to the log-in screen 3. System displays that the log-in failed and displays a link “Click here to Register” that will take the user to the Register page if click   **Alternate #2:** Doctor searches for a Patient that does not exist in the system   1. System take the Doctor back to previous screen 2. System displays a message that notifies the Doctor that the Patient doesn’t exist or has not yet registered for HealthNet |
| **Post Condition:** | Doctor is able to view the statistics of selected patient |

|  |  |
| --- | --- |
| **Use Case Number:** | **UC-20-01** |
| **Use Case Name:** | Password Recovery |
| **Overview:** | Any user type should have the ability to recover a lost or forgot password |
| **Actor(s):** | User (Patient, Doctor, Nurses, System Admins) |
| **Pre Condition(s):** | * System has been setup and configured * User has an account with the website * User has accessed website via URL |
| **Scenario Flow:** | 1. User Clicks log-on 2. User attempts *n* passwords (failing to log in) 3. User then clicks on recover password 4. User is asked to enter the user name that he need to recover pass word for 5. User is then prompted with questions about the account 6. If the answers are correct then the user is prompted to enter a new password 7. New Password is saved to the account |
| **Alternate Flows:** | **Alternate #1:** User is unable to answer the question correctly. User must then contact the system administer to unlock the account. |
| **Post Condition:** | User should now be able to log into there account. |

**5.0 System Requirements (Software Features)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements ID** | **Relevant Use Case Numbers** | **Description** | **Targeted Release** |
| REQ-001 | UC-01,  UC-02,  UC-03 | Create a user structure that can store necessary medical data. | R1 |
| REQ-002 | UC-02,  UC-04 | Create a hierarchy of permissions for types of users (Nurse, Doctor, and Administrator) allowing them to access and modify only the data that they can control. | R1/R2 |
| REQ-003 | UC-06,  UC-07,  UC-08 | Create a calendar for Doctor users that all patients of the Doctor can see.   * Allow all user types to create and update appointments * Allow nurses to modify existing appointments * Allow Doctors and patients to cancel their appointments | R2 |
| REQ-004 | UC-09,  UC-10,  UC-11,  UC-17 | Create a file database for more complex information such as x-rays, prescriptions, and other diagnostic tests  Allow doctors to update and upload tests, prescriptions, and other medical documents  Allow patients see released tests and medic documents | R2 |
| REQ-005 | UC-12,  UC-14 | Log important actions by all users with a user and time ID that admins can view at a later date. | R2 |
| REQ-006 | UC-13,  UC-16 | For each hospital create a log of patients currently at the hospital, if there for an extended stay make a note of their stay and what they needs to be done to the patient  Allow Doctors and Nurses to admit patients for extended stays  Allow only Doctors to discharge patients | R2 |
| REQ-007 | UC-15,  UC-19 | Access all of the patients medical records and anonymize them to follow privacy constraints. And display them to necessary staff. | R2 |
| REQ-008 | UC-18 | Allow Doctors and Nurses to communicate back and forth with patients on health care needs | R2 |

**6.0 Design Constraints**

The following sections identify special constraints placed by the customer. These can range from software and hardware limitations to requirements in deployment. Other constraints beyond the scope of these two will be listed in the *Other* *Constraints* section.

**6.1 Software Constraints**

The entire web application will be constructed using the Django framework. As such, expertise in the Python programming language is required. Version control system will be done through Subversion (SVN) and developers are expected to commit often and in an orderly fashion. The commits should include detailed commit messages that are helpful to all other developers. The directory should be kept clean and organized; containing only files that are relevant to the needs of the project.

**6.2 Deployment Constraints**

Besides being accessible from the Internet, the entire project must be compatible with the current versions of the RIT Software Engineering environment for deployments and demonstrations.

**6.3 Other Constraints**

The entire team is expected to schedule, plan, and show their progress to the consumer through an online board at the website www.trello.com. The team will record their contact information, meeting times, and a weekly status card that will include a weekly summary of the team’s high-level accomplishments.

**7.0 Project Rationale**

While we consider all use cases to be important, the following in the list did not make it to the final release due to the following circumstances:

UC - 5 Export Information:

Not implemented due time constraints and being a use case which the team decided was below the priority of the others.

UC - 18 Send Private Message:

Not implemented due time constraints and being a use case which the team decided was below the priority of the others, since this functionality is not unique to the system and there are plenty of other resources for sending online messages (e-mail, for example).

UC - 20 Password Recovery:

Not implemented due the high opportunity cost; the complexity of working out a secure hash function that would retrieve/generate a password made it more time-consuming to implement than other more important use cases.

**8.0 Document Change Log**

|  |  |  |  |
| --- | --- | --- | --- |
| ***Document Version Number*** | ***Revision Date*** | ***Summary of Changes*** | ***Author(s)f*** |
| *Initial / v0.0* | *02/09/2015* | *Initial revision.* | *Scott Baron, Martin Suarez, Zachary Glassner, Han Thu, Brian Mejorado* |
| *v0.1* | *03/08/2015* | *After looking over the requirements and the deadline for the Release 1, we have decided to push use cases UC-10, UC-11, UC-12, UC-14, and UC-17 to Release 2.* | *Brian Mejorado, Scott Baron* |
| *R1 / v0.2* | *03/9/2015* | *Changed the Use Case Diagram in Clause 4.1 to one that should suit the customer’s requirements.* | *Scott Baron* |
| *v1.01* | *03/03/2015* | *Updates the Releases; going through the product the true functionality of the site is not what we had first anticipated. Added the Project Rationale to Clause 7.0* | *Scott Baron* |
| *v1.1* | *03/18/2015* | *UC-05 now specifies csv file format availability. UC-15 is updated with all statistics that have been specified by the customer. Added a UC-20 for password recovery based on prior communications with the customer.* | *Brian Mejorado* |
| *v1.2* | *4/4/2015* | *UC-20 description is now updated with more information about the Use Case and the way it should work* | *Scott Baron* |
| *v1.3* | *4/5/2015* | *Updated the Release Dates in response to the changes we discussed in our team meeting on Friday* | *Scott Baron* |
| *v1.4* | *4/6/2015* | *Changed the wording of Project Rationale (Clause 7.0).* | *Scott Baron* |
| *v1.5* | *4/18/2015* | *Removed an alternative flow from UC-06-01 in Clause 4.2 that seemed not to apply to the use case. Split UC-10-01 in Clause 4.2 into UC-10-01, UC-10-02, and UC-10-03. Fixed spelling/grammar in Clause 8.0.* | *Brian Mejorado* |
| *v1.6* | *4/19/2015* | *After deliberating, we have decided to remove the requirement that the administrator password be generated by the system in the description of UC-02-01 in Clause 4.2 (reason: it was never specified by the customer).*  *Updated REQ-004 in Clause 5.0 to include UC-09 based off of the system requirement description.* | *Brian Mejorado* |
| *v1.7* | *4/20/2015* | *Fixed the numbering in the description for UC-18-01 in Clause 4.2.* | *Brian Mejorado* |
| *v1.8* | *4/22/2015* | *Removed redundant line in UC-07-01 in Clause 4.2. Fixed/rewrote UC-06-01 to put it more in line with the actual use case.* | *Brian Mejorado* |
| *v1.9* | *5/4/2015* | *Updated the Use Case descriptions to be accurate with actual work-flow of the code.* | *Scott Baron* |
| *v1.10* | *5/5/2015* | *Updated the Use Case releases to reflect what will not be completed in time* | *Scott Baron* |
| *v1.11* | *5/5/2015* | *Updated Use Case descriptions updated* | *Scott Baron* |
| *v1.12* | *5/6/2015* | *Update section 7.0 Project Rationale to include use cases 5, 18, and 20* | *Martin Suarez* |