|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Actor** | **Description of Use Case (s)** | **Use case** | **Pre-condition** | **Post-condition** | **Requirements** |
| <<user>>  Medical Staff | This use case entails all the actions involved in Logging a user into and out of the system  -----------------------------------------------  This use case entails all the actions associated with the addition of patient information | **Login**  Login() | User must be on LDAP | User Is logged into system |  |
| Logout() | User must be actively logged in | User Is logged out of system and is not allowed to access system until logged in again |
| **Patient Details**  addPatientDetails() | User must be logged in. | Patient details are added. |  |
| previewForm()  *(before Submission)* | Form must be completed | Display preview of patient’s form before submission. |  |
| submitForm() | Form must be completed | Submit form to database. |  |
| Admin  (extends user) |  | **PatientDetails**  updatePatientDetails() | Patient must exist. Patient form must be completed. | Patients’s old details are replaced with new ones. |  |
| viewPatientDetails() | Patient must exist. | Patient details are displayed on client browser. |  |
| queryAllPatientDetails() | Patient must exist | Specialized query result is displayed on client browser. |  |
| **Form**  addNewForm() | Must be admin, form must not exist already | New form is created. |  |
| viewForm() | Patient name must be provided and must also exist. | View a specific from for a patient. |  |
| removeForm() | Form must exist. No data must be present in the form. | Form is deleted |  |
| editFormFields() | Field must exist. |  |  |
| addFormFields() |  |  |  |
| deleteFormFields() |  |  |  |
| downloadPatientForms() | Patient’s information must exist. |  |  |
| downloadPatientForm(String form name) |  |  |  |
| **CRUD\_User**  addNewUser() |  |  |  |
| changeUserPermissions() |  |  |  |
| deleteUser() |  |  |  |
|  | This use case authorizes and permits the different types of users to perform actions according to their hierarchical level | **UserAthorization**  IdentifyMe() | User must be actively logged in | User is identified and user type is matched to identification |  |
|  |  | setMyPermissions() | User type must be correctly identified | Permissions are given to user according to type matched |  |
|  |  | getMyPermissions() | User permissions must be correctly set | Permissions are granted to user |  |