

需求可追踪查询工具

文件 工具 视图 帮助

UC18

搜索

The administrator has authenticated himself or herself in the iTrust Medical Records system (UC2).

18.2 Main Flow:

An administrator chooses to maintain the hospital listing [S1]. (Note: A personnel may be assigned to more than one (UC 2, S1))

18.3 Sub-flows:

[S1] The administrator will store (1) hospital ID number for the hospital [E1], (2) up to 30 alphanumeric characters giving the name of the hospital, and (3) an address for the hospital.

[S2]. The system shall enable the administrator to add a new entry for a hospital, or modify the hospital name in an existing entry. Note that the administrator is not allowed through

class	score
HospitalBeanValidator	0.5488264413403809
HospitalsDAO	0.5135814284430577
auth.admin.hospitalListin...	0.4927562617140615
UpdateHospitalListAction	0.4848293785586698
auth.surveyResults_jsp	0.24127666829677188
EditOfficeVisitAction	0.23006872008975304
PersonnelDAO	0.12121889211199564
OfficeVisitDAO	0.09172154734244109
auth.hcp-uap.editOfficeVi...	0.0781873368984254
SurveyResultDAO	0.0735006270910082
UpdateCPTCodeListAction	0.060110840902943496
RemoteMonitoringDAO	0.05526028854244778
auth.hcp.addUAP_jsp	0.050470063013117863
UpdateNDCodeListAction	0.046718017433207
ViewSurveyResultAction	0.04668926727998832
auth.admin.editLOINCCo...	0.04391889096183396
auth.admin.addHCP_jsp	0.04310139227459936
auth.admin.editCPTProce...	0.03907095728194449
auth.admin.editNDCInter...	0.03798110961500098
auth.hcp-uap.addPatient_...	0.03528396921886691
auth.hcp-uap.viewReport...	0.03491716224218462
auth.admin.editICDCode...	0.03469733102953379
auth.admin.editNDCodes...	0.03309814455779751
TransactionDAO	0.029467577308995855
PersonnelValidator	0.023656766374178146
UpdateICDCodeListAction	0.023369909964986786
LOINCBeanValidator	0.022337657916166598
PatientValidator	0.021560327691929455
LOINCDAO	0.02098637715724724
MedicationBeanValidator	0.020692417421543687

用户判断

文件 工具 视图 帮助

需求 UC18 类 HospitalsDAO

☐ 相关☐ 不相关☐ 跳过☐ 停止判断

帮助

提交

HospitalsDAO

```
* @author Andy
*
*/
public class HospitalsDAO {
    private DAOFactory factory;
    private HospitalBeanLoader hospitalLoader = new HospitalBeanLoader();

    /**
     * The typical constructor.
     * @param factory The {@link DAOFactory} associated with this DAO, which is
     * used for obtaining SQL connections, etc.
     */
    public HospitalsDAO(DAOFactory factory) {
        this.factory = factory;
    }

    /**
     * Returns a list of all hospitals sorted alphabetically
     *
     * @return A java.util.List of HospitalBeans.
     * @throws DBException
     */
    public List<HospitalBean> getAllHospitals() throws DBException {
        Connection conn = null;
        PreparedStatement ps = null;
        try {
            conn = factory.getConnection();
            ps = conn.prepareStatement("SELECT * FROM Hospitals ORDER
BY HospitalName");

            ResultSet rs = ps.executeQuery();
            return hospitalLoader.loadList(rs);
        } catch (SQLException e) {
            e.printStackTrace();
            throw new DBException(e);
        } finally {
            DBUtil.closeConnection(conn, ps);
        }
    }
}
```