厚生労働科学研究費補助金（医薬品・医療機器等レギュラトリーサイエンス総合研究）
分担研究年度終了報告書

医薬品が関連したヒヤリ・ハット事例報告システムの設計・開発

研究分担者　木村　昌臣　芝浦工業大学

研究要旨

調剤ヒヤリ･ハットおよび疑義照会の報告システムおよび実際の運用を想定した認証等に関わる仕組みについて必要となる入力方式およびデータベース構造の検討を行い、Web ベースの調剤ヒヤリ･ハット事例および疑義照会事例の報告システムとして必要な機能の設計およびそのプロトタイプシステムの開発を行った。

A. 研究目的

日本医療機能評価機構による薬局ヒヤリ・ハット収集・分析事業において収集・公開されたヒヤリ・ハット事例の分析により 報告者に対してより本質的な要因の報告を促す仕組みが必要であり、報告者が意図した事例内容・要因と合った選択肢を用意する必要があることがわかった。また、タブレット端末の普及から、タブレット端末による報告が増加することが期待されるため、本研究では本質的な要因を報告することが可能なシステムの仕様の策定を行い、そのプロトタイプを構築することを狙いとした。

初年度の研究では、調剤に関わるヒヤリ・ハット事例報告機能のプロトタイプを作成し、次年度の研究では調剤と並んで重要性が高い疑義照会事例報告機能についての検討を行った。

B. 研究方法

まず、日本医療機能評価機構による薬局ヒヤリ・ハット収集・分析事業において収集・公開されたヒヤリ・ハット事例の分析を行い、調剤ヒヤリ・ハット事例および疑義照会事例を収集する際に必要となる事例内容および要因を表す選択し項目についての検討を行った。

具体的には、調剤ヒヤリ・ハット事例の事例内容については、薬局ヒヤリ・ハット事例の事例内容に関する自由記述を、そこに含まれる単語の有無にもとづいてクラスタリング手法を用いて分類し、その要因については「ヒューマンエラーは人間がとった認識や行動などの誤りであり、本質的な要因はさらにその背後にある」という考え方にもとづき、Situation Awareness に関する Endsley のモデルにもとづいて整理を行った。

また、疑義照会事例報告に関しては、薬局

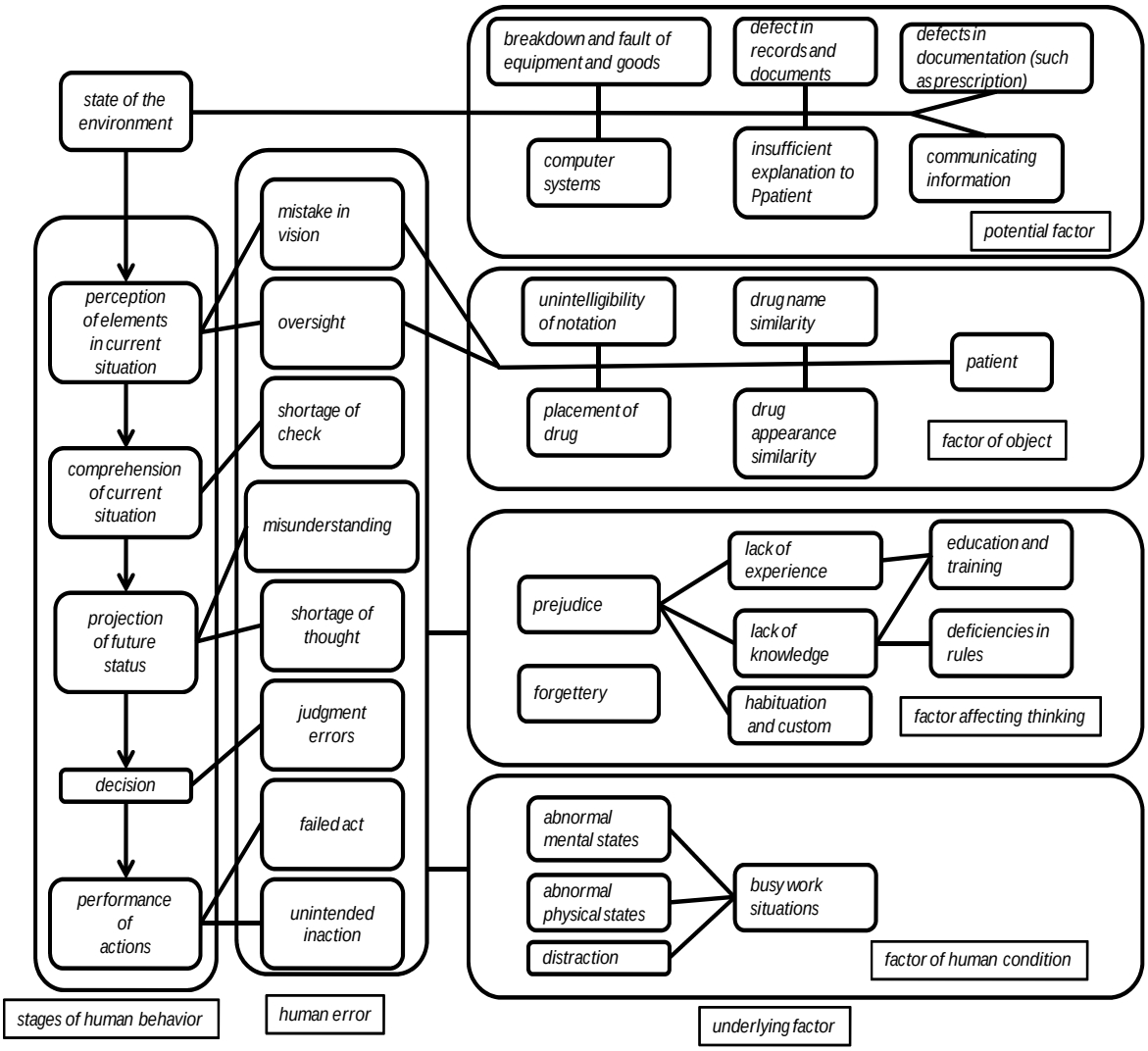

Figure 1　Endsley モデルによる要因・ヒューマンエラーの整理（調剤ヒヤリ・ハット）

ヒヤリ・ハット収集・分析事業にて現在公開されている疑義照会の事例内容として入力された自由記述文に単語間リンク法、格フレームによる分類などの手法を適用し、必要とされる項目についての検討を行った。

これを踏まえ、それぞれの結果で必要であるとわかった項目を入力可能なシステムの設計を行った。設計を行ったのは

- ユーザー管理
- 調剤ヒヤリ・ハット事例入力
- 疑義照会事例入力

である。

C. 研究結果

1. 調剤ヒヤリ・ハット事例入力

薬局ヒヤリ・ハット収集・分析事業にて公開されている事例内容データを分析したところ、「処方薬以外の調剤」「薬袋への入れ間違い」「数量間違い」等、21 種類に分類できることがわかった。また、Figure 1 に示した Endsley モデルにもとづく要因の整理により、ヒューマンエラーに関しては、

「見落とした」「見間違った」等、8 種類に分類され、要因について「心理的要因」「身体的要因」「慣れ・習慣」「医薬品外観類似」「医薬品名称類似」など 15 種類に分類されることがわかった。ただし、要因については、このままであると事例入力者が理解しにくく却って事実と異なる入力がなされる恐れがあるため、「焦り」「混雑」「作業手順の不履行」など業務に近い言葉による 16 種類の分類に置き換えた。

調剤ヒヤリ・ハット事例の入力に際しては、分析の結果得られた事例の内容に関する選択肢を選び、自由記述で詳細を報告者に記載してもらうことに加え、発生要因に関しては、まず発生したヒューマンエラーを先に選択し（ヒューマンエラーに関係ない場合は「その他」を選択）、その後に対応する背景要因を入力することができるユーザーインターフェイスとした。さらに、個々の背景要因について自由記述により詳細を記載できるようにした。

入力項目を下記に示す。

- 発生年月日
- 発見者
- 当事者
- 発生場面
- 事例の内容・自由記述
- 発生要因
  - ヒューマンエラーのカテゴリー
  - 背景要因・自由記述
- 処方された医薬品
  - HOT11
  - 販売名
  - 製造会社
  - 販売会社
- 間違えた医薬品
  - HOT11
  - 販売名
  - 製造会社
  - 販売会社

発生年月日は入力忘れ・入力ミスを防ぐため、入力フィールドもしくはボタンを押下するとカレンダーが表示され、これを選択することにより入力する方法をとった

これらの項目を入力するアプリケーションは Web アプリケーションとしてプロタイプを実装したが、その際、PC 端末ばかりでなくタブレット端末からの入力も想定して、事例内容入力については画面遷移を極力少なくするため Javascript のライブラリである jQuery を活用した。

2. 疑義照会事例入力

薬局ヒヤリ・ハット事例収集・分析事業で公開されている疑義照会事例の自由記述項目「事例内容」に入力されたデータ（自然言語文）を対象にどのような入力項目及びその選択肢が必要であるかの検討を行った。

単語間リンク法とは適用する全ての文章に含まれる係り受けのうちある一定頻度以上のものが構成するグラフ構造から頻出する文構造を発見する手法であるが、これを項目「事例内容」に適用したところ、「患者に＜薬剤＞が処方されたため、医師に疑義照会したところ、＜薬剤＞を＜用法・用量＞に変更した」という記述パターンが読み取れた。これにより、構造として「○○したため、疑義照会したところ△△」という部分は疑義照会の理由とその結果を示していると考えることができ、疑義照会を行った理由に相当する「疑義照会する」に係りか

つ「ため」で終わる文節と、疑義照会の結果を示している「疑義照会する」に係られる「ところ」にさらに係られる文節を抽出した。その結果、理由を示す「ため」が大きく分けて二つの使われ方をしていることがわかる。すなわち、「処方される」「多い」「少ない」「重複する」「併用禁忌」「異なる」など明らかな根拠を差しそれを疑義照会の理由にあげているものと、「確認する」「可能性がある」「考えられる」「思われる」など明確な根拠はないが疑わしい点があることが理由となっているものである。後者についての詳しい情報を得るために、特に動詞「確認する」に係る文節を助詞ごとにまとめて収集した。確認をする相手として医師、患者およびその家族、確認をする手段としてお薬手帳、薬歴、併用薬、処方箋などが挙げられる。また、疑義照会の結果、薬剤・用法・用量が追加・変更・削除となっている事例が多いことがわかる。以上の結果およびそれより類推される事柄を考慮し、項目として「疑義照会に至った理由」「判断のもとになる情報源」「疑義照会を行った結果」が必要であることがわかり、それぞれに対しての選択肢を決定した。

3. ユーザー管理

実際にシステムを運用する際に要となる

- 特権ユーザー管理
- 施設・薬局ごとの一般ユーザー管理および事例収集機能（特権ユーザー機能）
- 事例報告機能（一般ユーザー機能）

の機能についても設計を行った。

システムに対するセキュリティ要件として、

- 入力されたデータと入力した組織が入力後に紐付かないこと
- 同じ組織の安全管理責任者としての登録であっても、登録のタイミングにより安全管理責任者ユーザーを区別する仕組みであること

を考慮した。前者は、事例情報を登録するサーバーに対する不正アクセスが万が一発生した場合であっても、事例登録を行った組織にどのようなヒヤリ・ハット事例が生じていたかについて他者に漏洩されることを防ぐためのものである。そのため、登録後に各組織に対して組織内の一般ユーザーによりどのような事例報告がなされたかという情報を提供できない。そのため、一般ユーザーが報告した内容をメールにて安全管理責任者に送信することとした。また、本システムの設計にあたって、特に安全管理責任者の登録・認証が本来の責任者以外によって行われた場合であっても、一般ユーザーが入力した情報が影響を受けないよう、安全管理責任者に対するユーザー名、PIN、パスワードの三つ組みによる認証を提案した。

以上のデータを格納するべきデータベースについては、付録に掲載する。

D. 考察

調剤ヒヤリ・ハット事例については、事例に直接関係するヒューマンエラーとさらにそれが生じる本質的な要因を分離して報告できるシステムが実現可能であることを、プロトタイプを構築することにより示した。ヒューマンエラーは人間に起因するエラーであるが、さらにそれを誘引する疲れや業務多忙などの要因があることが考えられる

が、従来はこのふたつがフラットに考えられていた。現在でもヒューマンエラーを防ぐための試みは十分に実施されているものの完全になくすことは困難である。むしろ、ヒューマンエラーを起こす要因を抑制する方法があればそれによりヒューマンエラーを減らすことができ、ヒヤリ・ハット事例も減らすことができることが期待される。一方で、システムの入力者がヒューマンエラーとその要因を区別して入力ができるかについてはより一層の検討が必要である可能性がある。なぜならば、ヒューマンエラーは実際に発生した事例で目に見えやすく認知しやすいエラーであるのに対し、発生要因はヒューマンエラー発生の根本原因がなんであるかという検討が不可欠であり、そのため、入力者によっては従来の入力方法と比べて時間がかかる可能性があるからである。しかしながら、上に述べた理由で本質的な要因を明らかにすることは非常に重要であるため、実際の入力のされ方をもとに文言の修正・追加等による改善は絶えず行っていく必要があると考えられる。

また、疑義照会事例の入力項目としては、薬局ヒヤリ・ハット収集・分析事業において収集・公開されたヒヤリ・ハット事例をテキストマイニングの手法をもとに解析し、決定した。これにより、実際に入力されることが多い項目・選択肢が選択されたと考えられるが、その一方で、テキストマイニングという手法の特性上、頻度が多い事例を中心とした選択のされ方となっている。そのため、頻度は少ないが見過ごすと被害の程度が大きいものについても今後追加することを検討することが望ましい。

ユーザー管理については、システム内部でユーザー名と PIN の組が安全管理責任者と一対一対応するよう設計したが、安全管理責任者の視点ではユーザー名に対してパスワード相当のものが二つあるように捉えられることができる。PIN もパスワードもセキュリティ上、他人に知られないよう管理する必要があるものであるので、本システムの仕組みを利用するために新しい概念を導入する必要がなく、4 桁の数を追加で管理する必要はあるものの安全管理責任者に対して大きな負担をかけずに利用が可能であると考えられる。

E. 結論

本研究では調剤時のヒヤリ・ハット事例および疑義紹介の事例を収集するシステムを設計・開発した。これにより、以下の仕組みが必要であることがわかった。

- 事例を入力する一般ユーザーと、自組織の一般ユーザーおよび入力された事例報告を管理する特権ユーザー（安全管理責任者）が必要である。
- 成りすましを目的として同一組織の安全管理責任者が複数登録されることによる情報漏えいを防ぐための認証方式が必要である。本システムではユーザー名、PIN、パスワードの組による認証を提案・実現した。
- 万が一、情報が漏洩した場合でも、入力されたデータと入力した組織が入力後に紐付くことによる不利が生じない仕組みが必要である。本システムでは、データベース内部には登録者の情報を持たず、その代わりに報告は安全管理責任者にメールで送信される方法をとった。

- 調剤ヒヤリ・ハット事例については、事例に直接関係するヒューマンエラーとさらにそれが生じる本質的な要因を分離して報告できるシステムが実現可能であることを、プロトタイプを構築することにより示した。
- 疑義照会事例の報告については、薬局ヒヤリ・ハット収集・分析事業において収集・公開されたヒヤリ・ハット事例を解析したことにより、疑義照会を行う必要があると判断した理由、疑義照会を行う必要があるかどうかを確認するための情報源、疑義照会を行ったことによって生じた結果のそれぞれの情報を収集する必要があることがわかった。

今後は、特にタブレット端末等における既存システムのユーザーインターフェイスとの使いやすさ・入力効率等の観点からの比較を行う必要がある。さらに、実際に本システムを運用した上で、入力項目等の精査も今後行う必要がある。

F．論文発表

[1] 佐藤隆亮, 鍋田啓太, 木村昌臣, 大倉典子, 土屋文人: “薬局ヒヤリ・ハット事例の解析”, 電子情報通信学会 2011 年ソサエティ大会講演論文集

[2] 佐藤隆亮, 鍋田啓太, 木村昌臣, 大倉典子, 土屋文人: “薬局ヒヤリ・ハット事例の解析(第二報)”, 電子情報通信学会 2012 年総合大会講演論文集

[3] 佐藤隆亮, 鍋田啓太, 木村昌臣, 大倉典子, 土屋文人: “薬局ヒヤリ・ハット事例の解析(第三報)”, 電子情報通信学会 2012 年ソサエティ大会講演論文集

[4] T. Sato, R. Okuya, M. Kimura, M. Ohkura, and F. Tsuchiya, "Analysis on Incident Data in Pharmacies," International Journal of Computer and Electrical Engineering vol. 5, no. 2, pp. 246-250, 2013.

[5] 佐藤隆亮,木村昌臣, 大倉典子, 土屋文人: “薬局ヒヤリ・ハット事例の解析(第四報)”, 人間工学 Vol. 49 No. Supplement p. S276-S277 (2013)

[6] 佐藤隆亮, 木村昌臣, 大倉典子, 土屋文人: “薬局ヒヤリ・ハット事例の解析(第五報)”, 電子情報通信学会 2014 年総合大会講演論文集

[7] T. Sato, M. Kimura, M. Ohkura, and F. Tsuchiya, "Analysis on Incident Data in Pharmacies (II)" Proceedings of CETC2013, #77, Lisbon, Portugal, 2013.

付録：

- **システムのデータベース定義**

```
CREATE DATABASE  IF NOT EXISTS `incidentdb3` /*!40100 DEFAULT
CHARACTER SET utf8 */;
USE `incidentdb3`;
--
-- Table structure for table `t_caseinfo`
--
DROP TABLE IF EXISTS `t_caseinfo`;
CREATE TABLE `t_caseinfo` (
  `CASEID` bigint(20) NOT NULL,
  `CASEDATE` date DEFAULT NULL,
  `DETECTOR` varchar(45) DEFAULT NULL,
  `PLAYER` varchar(45) DEFAULT NULL,
  `SCENE` varchar(45) DEFAULT NULL,
  `CONTENT` varchar(45) DEFAULT NULL,
  `CONTENTTEXT` text,
  `INPUTUSERID` varchar(45) NOT NULL,
  `PCODE` varchar(9) NOT NULL,
  `INPUTDATE` datetime NOT NULL,
  PRIMARY KEY (`CASEID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
-- Table structure for table `t_count`
--
DROP TABLE IF EXISTS `t_count`;
CREATE TABLE `t_count` (
  `COUNT` bigint(20) NOT NULL,
  PRIMARY KEY (`COUNT`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
--
-- Table structure for table `t_drug`
--

DROP TABLE IF EXISTS `t_drug`;
CREATE TABLE `t_drug` (
  `HOT11` varchar(11) NOT NULL,
  `DRUG_NAME` varchar(200) NOT NULL,
  `MAKER_NAME` varchar(45) DEFAULT NULL,
  `COMPANY_NAME` varchar(45) DEFAULT NULL,
  `DRUGTYPE` varchar(10) DEFAULT NULL,
  `GENERIC` varchar(10) DEFAULT NULL,
  PRIMARY KEY (`HOT11`),
  KEY `idx_drugtype` (`DRUGTYPE`),
  KEY `idx_name` (`DRUG_NAME`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
-- Table structure for table `t_pharmacy`
--

DROP TABLE IF EXISTS `t_pharmacy`;
CREATE TABLE `t_pharmacy` (
  `PCODE` varchar(9) NOT NULL,
  `PREFCODE` varchar(2) DEFAULT NULL,
  `PREFECTURE` varchar(10) DEFAULT NULL,
  `PHARMACY_NAME` varchar(45) DEFAULT NULL,
  `ADDRESS` varchar(200) DEFAULT NULL,
  `TEL` varchar(20) DEFAULT NULL,
  `PID` varchar(7) NOT NULL,
  PRIMARY KEY (`PCODE`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
--
-- Table structure for table `t_presc_drug`
--

DROP TABLE IF EXISTS `t_presc_drug`;
CREATE TABLE `t_presc_drug` (
  `CASEID` bigint(20) NOT NULL,
  `NO` int(11) NOT NULL,
  `HOT11` varchar(11) DEFAULT NULL,
  `DRUGNAME` varchar(100) DEFAULT NULL,
  `MAKER` varchar(100) DEFAULT NULL,
  `COMPANY` varchar(100) DEFAULT NULL,
  `INPUTUSERID` varchar(45) DEFAULT NULL,
  `INPUTDATE` datetime DEFAULT NULL,
  PRIMARY KEY (`CASEID`,`NO`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;


--
-- Table structure for table `t_wrong_drug`
--

DROP TABLE IF EXISTS `t_wrong_drug`;
CREATE TABLE `t_wrong_drug` (
  `CASEID` bigint(20) NOT NULL,
  `NO` int(11) NOT NULL,
  `RIGHTNO` int(11) DEFAULT NULL,
  `RIGHTHOT11` varchar(11) DEFAULT NULL,
  `WRONGHOT11` varchar(11) DEFAULT NULL,
  `WRONGDRUGNAME` varchar(100) DEFAULT NULL,
  `WRONGMAKER` varchar(100) DEFAULT NULL,
  `WRONGCOMPANY` varchar(100) DEFAULT NULL,
  `INPUTUSERID` varchar(45) DEFAULT NULL,
  `INPUTDATE` datetime DEFAULT NULL,
  PRIMARY KEY (`CASEID`,`NO`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
--
-- Final view structure for view `v_drug`
--

DROP VIEW IF EXISTS `v_drug`;
CREATE VIEW `v_drug` AS select `a`.`HOT11` AS `HOT11`,`a`.`DRUG_NAME` AS
`DRUG_NAME`,`a`.`MAKER_NAME` AS `MAKER_NAME`,`a`.`COMPANY_NAME`
AS `COMPANY_NAME`,`a`.`DRUGTYPE` AS `DRUGTYPE`,`a`.`GENERIC` AS
`GENERIC`,`b`.`PACKAGE_FORM` AS
`PACKAGE_FORM`,`b`.`PACKAGE_NUMBER` AS
`PACKAGE_NUMBER`,`b`.`PACKAGE_UNIT` AS `PACKAGE_UNIT` from (`t_drug`
`a` join `t_package` `b`) where (`a`.`HOT11` = `b`.`HOT11`) */;

--
-- Table structure for table `drugname`
--

DROP TABLE IF EXISTS `drugname`;

CREATE TABLE `drugname` (
  `DRUGNAME` varchar(250) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
-- Table structure for table `inputdata3`
--

DROP TABLE IF EXISTS `inputdata3`;
CREATE TABLE `inputdata3` (
  `USER` varchar(12) DEFAULT NULL,
  `ID` int(4) DEFAULT NULL,
  `REASON1` tinyint(1) DEFAULT NULL,
  `REASON2` tinyint(1) DEFAULT NULL,
  `REASON3` tinyint(1) DEFAULT NULL,
  `REASON4` tinyint(1) DEFAULT NULL,
  `REASON5` tinyint(1) DEFAULT NULL,
```

```
  `REASON6` tinyint(1) DEFAULT NULL,
  `REASON7` tinyint(1) DEFAULT NULL,
  `REASON8` tinyint(1) DEFAULT NULL,
  `REASON9` tinyint(1) DEFAULT NULL,
  `REASON10` tinyint(1) DEFAULT NULL,
  `REASON11` tinyint(1) DEFAULT NULL,
  `REASON12` tinyint(1) DEFAULT NULL,
  `REASON13` tinyint(1) DEFAULT NULL,
  `REASON14` tinyint(1) DEFAULT NULL,
  `REASON15` tinyint(1) DEFAULT NULL,
  `REASONTXT` varchar(500) DEFAULT NULL,
  `SOURCE1` tinyint(1) DEFAULT NULL,
  `SOURCE2` tinyint(1) DEFAULT NULL,
  `SOURCE3` tinyint(1) DEFAULT NULL,
  `SOURCE4` tinyint(1) DEFAULT NULL,
  `SOURCE5` tinyint(1) DEFAULT NULL,
  `SOURCE6` tinyint(1) DEFAULT NULL,
  `SOURCE7` tinyint(1) DEFAULT NULL,
  `SOURCTXT` varchar(500) DEFAULT NULL,
  `RESULT1` tinyint(1) DEFAULT NULL,
  `RESULT2` tinyint(1) DEFAULT NULL,
  `RESULT3` tinyint(1) DEFAULT NULL,
  `RESULT4` tinyint(1) DEFAULT NULL,
  `RESULT5` tinyint(1) DEFAULT NULL,
  `RESULT6` tinyint(1) DEFAULT NULL,
  `RESULT7` tinyint(1) DEFAULT NULL,
  `RESULT8` tinyint(1) DEFAULT NULL,
  `RESULTTXT` varchar(500) DEFAULT NULL,
  `SOURCENOTE` varchar(500) DEFAULT NULL,
  `RESULTNOTE` varchar(500) DEFAULT NULL,
  `TEXTAREA` varchar(1000) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
--
-- Table structure for table `inputdrugs`
--

DROP TABLE IF EXISTS `inputdrugs`;
CREATE TABLE `inputdrugs` (
  `USER` varchar(12) DEFAULT NULL,
  `ID` int(4) DEFAULT NULL,
  `TYPE` varchar(10) DEFAULT NULL,
  `NUMBER` int(3) DEFAULT NULL,
  `DRUGNAME` varchar(200) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```