**CS673F13 Software Engineering**

**Group Project - Type II Diabetes Management CDSS**

**Software Design Document**



|  |  |  |  |
| --- | --- | --- | --- |
| Team Member | Role(s) | Signature | Date |
| Bogdan Chayka | Backup project leader, Design leader | *Bogdan Chayka* | 12/7/2013 |
| Jason Lu | Configuration leader |  | 12/7/2013 |
| Jeff Andre | Project leader, Requirements | *Jeff Andre* | 12/7/2013 |
| Maura Huff | Implementation leader |  | 12/7/2013 |
| Wenjie Shi(Jenny) | Environment and integration leader |  | 12/7/2013 |
| Yike Xue (Eva) | QA leader | *Yike Xue* | 12/7/2013 |
| Yingyuan (Allen) Zhang | Requirements leader | *Yingyuan Zhang* | 12/7/2013 |
| Yulu Liu | Design leader |  | 12/7/2013 |
| Yehuan Huang (Tommy) | Developer |  | 12/7/2013 |

**Revision history**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Author** | **Date** | **Change** |
| **1.0** | **Jeff Andre, Bogdan Chayka** | 11/07/2013 |  |
| **2.0** | **Yike Xue**  **Jeff Andre**  **Bogdan Chayka** | 12/7/2013 | **Model(database)**  **Design Patterns**  **Algorithm**  **Database Schema** |

[Introduction](#h.87t9hln2vjz0)

[Software Architecture](#h.buttcq9i221r)

[Design Patterns](#h.x18fj36s1121)

[Key Algorithms](#h.mtfbusfb0eq3)

[Classes and Methods](#h.7ucksmkf6rzx)

[References](#h.15tmymhipvdv)

[Glossary](#h.8n34lvocupub)

# Introduction

This web application is designed to be a teaching tool that demonstrate implementation of a real clinical algorithm (part of [AACE Diabetes Management Algorithm 2013](https://www.aace.com/files/aace_algorithm.pdf)), examples of successful treatment decisions, situations where clinical algorithms may fail, and to provide a platform for familiarization of students with the process of development and implementation and use of medical algorithms.

This application is divided into five major parts:

a) user interface

b) a database

c) a type II diabetes management algorithm

d) health calculators

e) security

Figure 1 shows a diagram of the application.

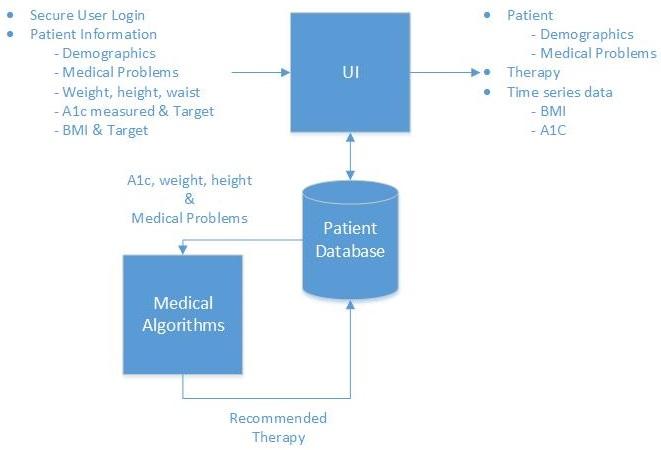


Figure 1: Diagram of the application

# Software Architecture

The [Model View Controller](http://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller) (MVC) architecture along with the [cakePHP](http://cakephp.org/) and [Bootstrap](http://getbootstrap.com/) frameworks, were used for development of this application. Figure 1 below shows a diagram of the whole MVC architecture with each component.

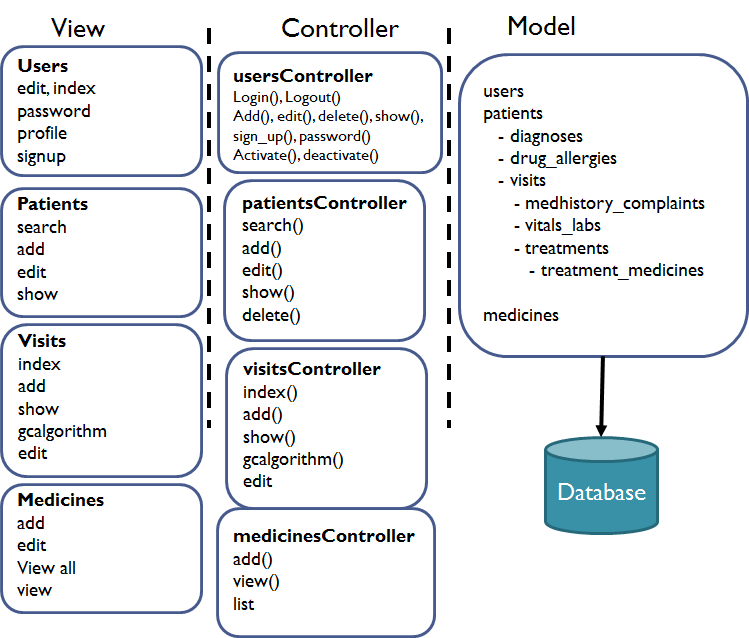


Figure 1. Architecture Diagram

**1. Model**

1.1 Database

[MySQL](http://www.mysql.com/) is used to implement the database and CakePHP naming conventions to name tables. Database design supports workflow of medical encounter. Close attention paid to security of database during designing, development and implementation. Enum type is widely used to strictly constrain the input and make sure the data stored in database is valid.

Due to the possible extension of the project, a few attributes in tables are not used and were designed for future development as well as diagnoses table.

Figure 3 shows a database schema.

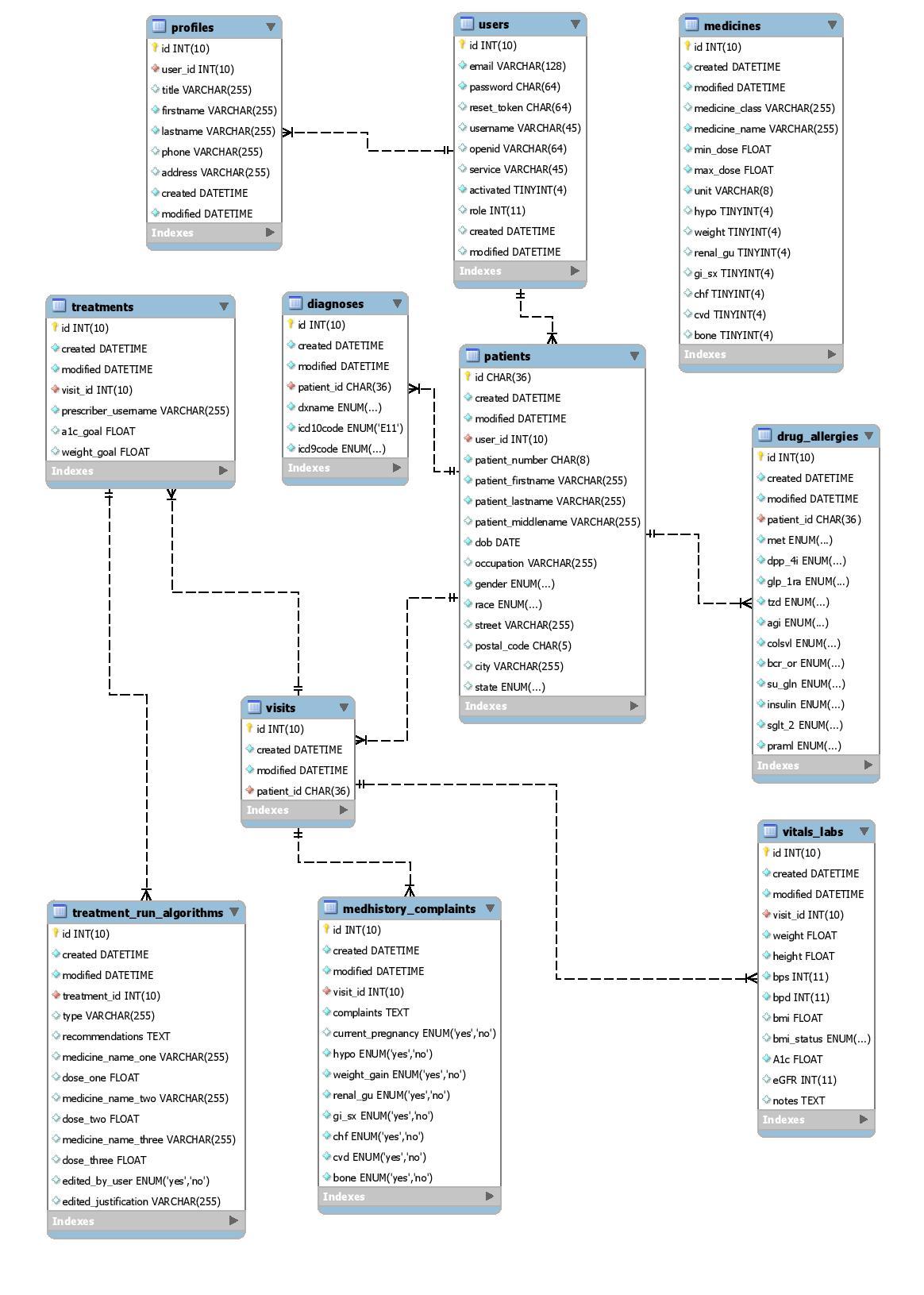


Figure 3: Database schema

Detailed descriptions of tables:

patients

|  |  |  |  |
| --- | --- | --- | --- |
| **attributes** | **types** | **constraints** | **keys** |
| id | char (36) | not null | PK |
| created | datetime | not null |  |
| modified | datetime | not null |  |
| user\_id | int | not null | FK |
| patient\_number | varchar(8) | not null |  |
| patient\_firstname | varchar(255) | not null |  |
| patient\_lastname | varchar(255) | not null |  |
| patient\_middlename | varchar(255) |  |  |
| dob | date | not null |  |
| picture | blob |  |  |
| occupation | varchar(255) |  |  |
| gender | enum | not null  ‘Male’, ‘Female’ |  |
| race | enum | not null  “African or African American”,  “Asian or Asian American”,  “Caucasian or European American”,  “Native American or Native Alaskan ”,  “Other Race” |  |
| street | varchar(255) |  |  |
| postal\_code | char(5) |  |  |
| city | varchar(255) |  |  |
| state | enum | All states  (see notes) |  |

visits

|  |  |  |  |
| --- | --- | --- | --- |
| **attributes** | **types** | **constraints** | **keys** |
| id | int | not null | PK |
| created | datetime | not null |  |
| modified | datetime | not null |  |
| patient\_id | char(36) | not null | FK |

medhistory\_complaints

|  |  |  |  |
| --- | --- | --- | --- |
| **attributes** | **types** | **constraints** | **keys** |
| id | int | not null | PK |
| created | datetime | not null |  |
| modified | datetime | not null |  |
| visit\_id | int | not null | FK |
| complaints | text | not null |  |
| hypo | enum | not null  “yes” or “no” |  |
| weight\_gain | enum | not null  “yes” or “no” |  |
| renal\_gu | enum | not null  “yes” or “no” |  |
| gi\_sx | enum | not null  “yes” or “no” |  |
| chf | enum | not null  “yes” or “no” |  |
| cvd | enum | not null  “yes” or “no” |  |
| bone | enum | not null  “yes” or “no” |  |

vitals\_labs

|  |  |  |  |
| --- | --- | --- | --- |
| **attributes** | **types** | **constraints** | **keys** |
| id | int | not null | PK |
| created | datetime | not null |  |
| modified | datetime | not null |  |
| visit\_id | int | not null | FK |
| weight | float | not null |  |
| height | float | not null |  |
| bps  designed for future  development | int | not null |  |
| bpd  designed for future  development | int | not null |  |
| bmi | float |  |  |
| bmi\_status | varchar(255) |  |  |
| A1c | float | not null |  |
| eGFR  designed for future  development | int |  |  |
| notes | text |  |  |

diagnoses

designed for future development

|  |  |  |  |
| --- | --- | --- | --- |
| **attributes** | **types** | **constraints** | **keys** |
| id | int | not null | PK |
| created | datetime | not null |  |
| modified | datetime | not null |  |
| patient\_id | char(36) | not null | FK |
| dxname | enum | ‘Non-insulin-dependent diabetes mellitus’ |  |
| icd10code | enum | not null,  “E11” |  |
| Icd9code | enum | Not null,  “250.00”, “250.02” |  |

drug\_allergies

|  |  |  |  |
| --- | --- | --- | --- |
| **attributes** | **types** | **constraints** | **keys** |
| id | int | not null | PK |
| created | datetime | not null |  |
| modified | datetime | not null |  |
| patient\_id | char(36) | not null | FK |
| met | enum | not null,  'yes','unknown','NKDA' |  |
| dpp\_4i | enum | not null,  'yes','unknown','NKDA' |  |
| glp\_1ra | enum | not null,  'yes','unknown','NKDA' |  |
| tzd | enum | not null,  'yes','unknown','NKDA' |  |
| agi | enum | not null,  'yes','unknown','NKDA' |  |
| colsvl | enum | not null,  'yes','unknown','NKDA' |  |
| bcr\_or | enum | not null,  'yes','unknown','NKDA' |  |
| su\_gln | enum | not null,  'yes','unknown','NKDA' |  |
| insulin | enum | not null,  'yes','unknown','NKDA' |  |
| sglt\_2 | enum | not null,  'yes','unknown','NKDA' |  |
| praml | enum | not null,  'yes','unknown','NKDA' |  |

treatments

|  |  |  |  |
| --- | --- | --- | --- |
| **attributes** | **types** | **constraints** | **keys** |
| id | int | not null | PK |
| created | datetime | not null |  |
| modified | datetime | not null |  |
| visit\_id | int | not null | FK |
| prescriber\_username | varchar(255) | not null |  |
| a1c\_goal | float |  |  |
| weight\_goal | float |  |  |

treatment\_run\_algorithms

|  |  |  |  |
| --- | --- | --- | --- |
| **attributes** | **types** | **constraints** | **keys** |
| id | int | not null | PK |
| created | datetime | not null |  |
| modified | datetime | not null |  |
| treatment\_id | int | not null | FK |
| type | enum | not null  “lifestyle modification”  “monotherapy”,  “dual\_therapy”,  “triple\_therapy” |  |
| recommendations | text |  |  |
| medicine\_name\_one | varchar(255) |  |  |
| dose\_one | float |  |  |
| medicine\_name\_two | varchar(255) |  |  |
| dose\_two | float |  |  |
| medicine\_name\_three | varchar(255) |  |  |
| dose\_three | float |  |  |
| edited\_by\_user | enum | not null “yes”, ”no” |  |

medicines

|  |  |  |  |
| --- | --- | --- | --- |
| **attributes** | **types** | **constraints** | **keys** |
| id | int | not null | PK |
| created | datetime | not null |  |
| modified | datetime | not null |  |
| medicine\_class  designed for future  development | varchar(255) | not null |  |
| medicine\_name | varchar(255) | not null |  |
| min\_dose | float | not null |  |
| max\_dose | float | not null |  |
| unit | varchar(8) | not null |  |
| hypo | tinyint(4) |  |  |
| weight | tinyint(4) |  |  |
| renal\_gu | tinyint(4) |  |  |
| gi\_sx | tinyint(4) |  |  |
| chf | tinyint(4) |  |  |
| cvd | tinyint(4) |  |  |
| bone | tinyint(4) |  |  |

users

|  |  |  |  |
| --- | --- | --- | --- |
| **attributes** | **types** | **constraints** | **keys** |
| id | int | not null | PK |
| created | datetime | not null |  |
| modified | datetime | not null |  |
| email | varchar(128) | not null |  |
| password | char(64) | not null |  |
| reset\_token | char(64) |  |  |
| username | varchar(45) |  |  |
| openid | varchar(64) |  |  |
| service | varchar(45) |  |  |
| activated | tinyint | not null  default 0 |  |
| role | int |  |  |

user\_profiles

|  |  |  |  |
| --- | --- | --- | --- |
| **attributes** | **types** | **constraints** | **keys** |
| id | int | not null | PK |
| created | datetime | not null |  |
| modified | datetime | not null |  |
| user\_id | int | not null | FK |
| title | varchar(255) |  |  |
| firstname | varchar(255) | not null |  |
| lastname | varchar(255) | not null |  |
| phone | varchar(255) |  |  |
| address | varchar(255) |  |  |

**1.2 Model**

* users
* patients
* diagnoses
* drug\_allergies
* visits
* medhistory\_complaints
* vitals\_labs
* treatments
* treatments\_medicines

**Glossary for database**:

Hypo – hypoglycemia

Weight – weight gain

Renal/GU Genitourinary and Renal symptoms

GI Sx – Gastrointestinal symptoms

CHF – coronary heart disease ( ischemic heart disease IHD)

CVD – cardiovascular disease

BONE – osteoporosis

eGFR – estimated glomerular filtration rate

a1c - blood glucose test

**Notes:**

Meaning of enum In medicines table

“0” - neutral

“1” - few adverse events or possible benefits

“2” – use with caution

“3” – likelihood of adverse effects

States in patients table:

'Alabama','Alaska','Arizona','Arkansas','California','Colorado','Connecticut','Delaware','Florida','Georgia','Hawaii','Idaho','Illinois','Indiana','Iowa','Kansas','Kentucky','Louisiana','Maine','Maryland','Massachusetts','Michigan','Minnesota','Mississippi','Missouri','Montana','Nebraska','Nevada','New Hampshire','New Jersey','New Mexico','New York','North Carolina','North Dakota','Ohio','Oklahoma','Oregon','Pennsylvania','Rhode Island','South Carolina','South Dakota','Tennessee','Texas','Utah','Vermont','Virginia','Washington','West Virginia','Wisconsin','Wyoming'

2. View Part

List of views (pages) for each object.

**Users:**

* edit
* index
* signup
* profile
* password

**Patients:**

* search
* add
* edit
* show
* index

**Visits:**

* add
* current
* show
* edit
* gcalgorithm
* index

**Medicines:**

* add
* edit
* view all
* view

3. Controller Part

List functions for each object.

**usersController:**

* login()
* index()
* do\_login()
* logout()
* add()
* edit()
* delete()
* show()
* sign\_up()
* password()
* activate()
* deactivate()
* profile()
* change\_password()
* new\_user()

**patientsController:**

* search()
* add()
* edit()
* show()
* delete()
* set\_patient\_id()
* index()
* admin()
* get\_a1c\_history()
* get\_bmi\_history()

**visitsController:**

* index()
* add()
* \_add()
* current()
* show()
* gcalgorithm()
* edit()

**medicinesController:**

* add()
* index()
* edit()
* delete()
* show()

# Design Patterns

* **Model - View - Controller (MVC) software pattern**

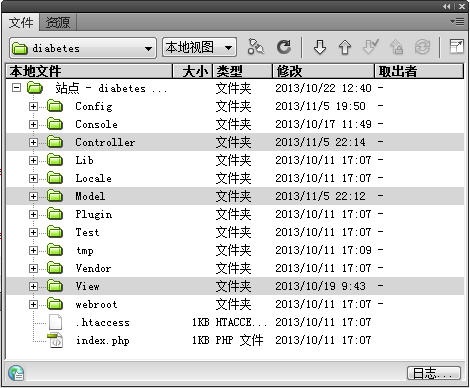
The application uses [MVC](http://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller) software pattern.

MVC separates the business logic and application data from the presentation data to the user. It consists of three parts: Model, View and Controller.

The model represents the business or domain logic of the application, typically with data backed by a database.

The view is what shows to the users. It is selected by the controller and renders the appropriate user interface.

The model is the core of this architecture. It links the model and view. The controller locates the appropriate action method in the controller, gets values to use as the action method's arguments, and handles any errors that might occur when the action method runs. It then renders the requested view.



* **Test Driven Development(TDD) design pattern**

Members would test some special cases and make them fail, then we try to fix these bugs and make the test cases passed. According to the test results, we can find out where we should improve and modify. clearly After the tests, we would modify our code until the tests pass. That is how our software getting improved.

The workflow is as following:

1. Create a test case based on the requirements and make it failed.

2. Write code to pass the test in Step1.

3. Refactor so that the code is made to be readable, maintainable and make sure it can still pass the test.

4. Repeat for each new test until all the requirements are met.

Besides, we also apply TDD to algorithm development using PHPUnit and Xdebug. There is one goal -- to make all the red turn green.

# Key Algorithms

A key algorithm used in this project is a part of the AACE Diabetes Management Algorithm 20131 which focuses on glycemic control of patients with type 2 diabetes mellitus. The two figures below show flow diagrams of the algorithm. Figure 4 shows the the initial loading of patient data and medicine tables from the database. Figure 5 shows the overall flow diagram for therapy selection based on A1c levels, and figure 6 shows the flow diagram for medication selection for each therapy. The algorithm only recommends a therapy for type 2 diabetes patients which includes therapeutic lifestyle changes plus one, two, or three medications - referred to as monotherapy, dual therapy and triple therapy respectively. Finally, figure 7 shows the flow diagram to check for patient risk of side effects for each selected medicine.

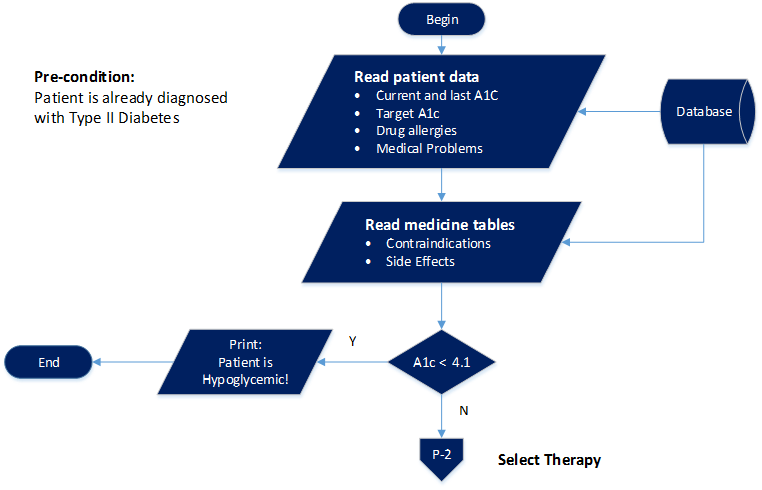


Figure 4. Read Patient data and Medicine tables from database

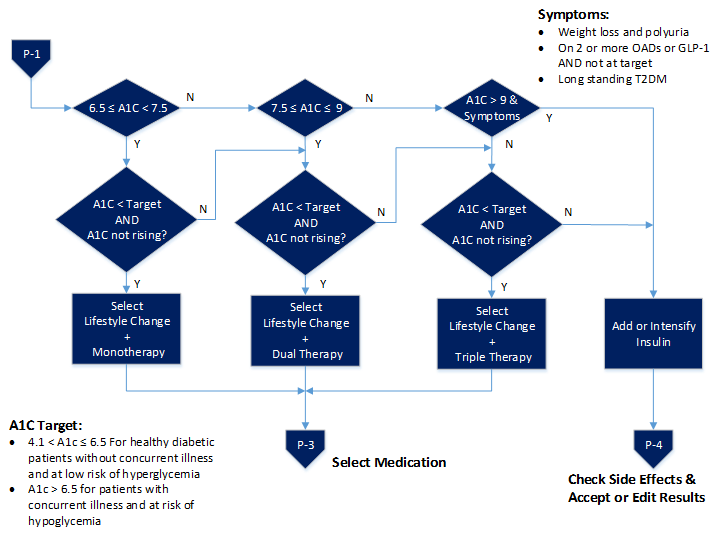


Figure 5. Monotherapy, Dual Therapy, & Triple Therapy flow diagram

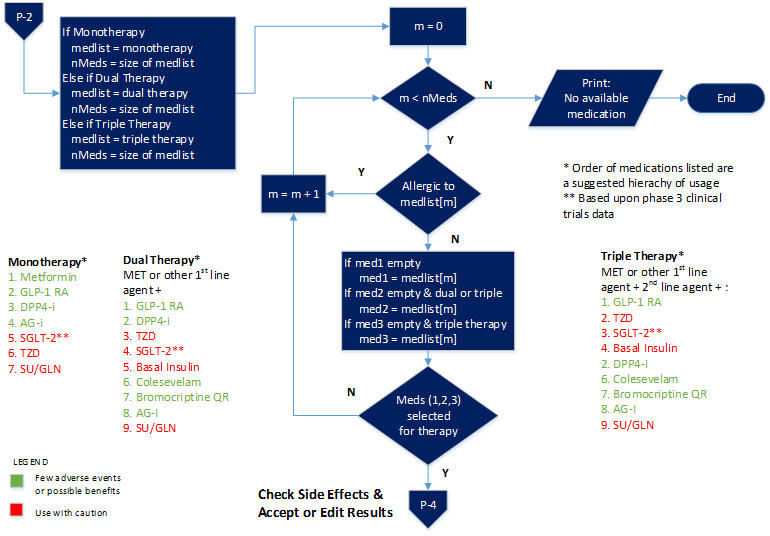


Figure 6. Medicine Selection

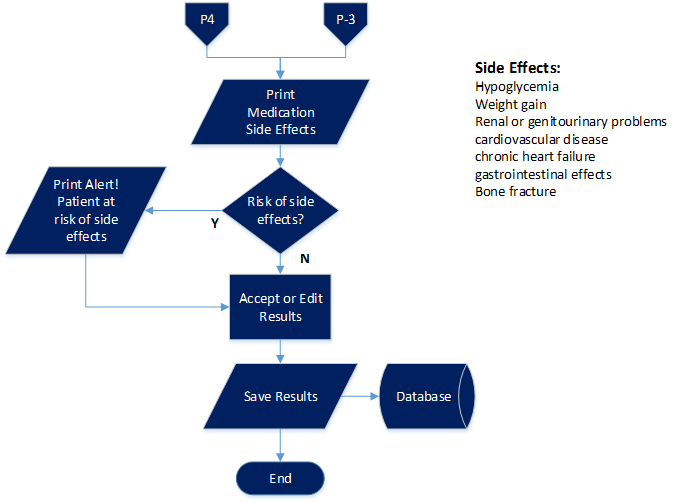


Figure 7. Check for Patient Risk of Side Effects

# Classes and Methods

This part can be a reference to automatic generated document for all classes and methods.

The documents in this section were automatically generated by phpDocumentor and can be viewed by clicking on the index.html file in a browser to see classes and methods in the following directory:

[github dir]\docs\phpdocs\index.html

**Classes:**

usersController, patientsController, visitsController, medicinesController

**Methods:**

usersController:

* login()
* index()
* do\_login()
* logout()
* add()
* edit()
* delete()
* show()
* sign\_up()
* password()
* activate()
* deactivate()
* profile()
* change\_password()
* new\_user()

patientsController:

* search()
* add()
* edit()
* show()
* delete()
* set\_patient\_id()
* index()
* admin()
* get\_a1c\_history()
* get\_bmi\_history()

visitsController:

* index()
* add()
* \_add()
* current()
* show()
* gcalgorithm()
* edit()

medicinesController:

* add()
* index()
* edit()
* delete()
* show()

# References

[1] AACE Comprehensive Diabetes Management, *Endocr Pract.* 2013;19(Suppl 2).

<https://www.aace.com/files/aace_algorithm.pdf>

# Glossary

AACE - American Association of Clinical Endocrinologists

A1c - Common blood test used to diagnose and manage type 1 and type 2 diabetes.

Also, known as glycated hemoglobin, glycosylated hemoglobin, hemoglobin A1C and

HbA1c.

BONE – osteoporosis

CHF – coronary heart disease ( ischemic heart disease IHD)

CVD – cardiovascular disease

GI Sx – Gastrointestinal symptoms

Hypo – hypoglycemia

Renal/GU Genitourinary and Renal symptoms

Weight – weight gain