Software Design Document

CS308: Large Applications Practicum
Indian Institute of Technology Mandi

October-November 2019

Group 9

B17096	Namrata Malkani
B17031	Aashima
B17033	Akhil Rajput
B17064	Suraj Kumar

Time Table Assist Tool
Design Document

Revision History

Version	Date	Author(s)	Description
v1.0	15/10/19	Namrata Malkani	Initial version
V2.0	06/11/19	Aashima	Final Version

Table of Contents

1 Introduction

Time-Table Assist Tool is a user interface designed to prevent errors while making time-table. It does not make the time table, it just assists the user by flashing errors. The user interface is connected to a database. The database stores all the information required to evaluate clashes. Each course is to be allotted a slot in the time table, a classroom and instructor(s). The entire decision of allotment lies with the user but the interface warns him from making clashes. Therefore, data entries violating constraints are not inserted into the time table and user is flashed error so that he/ she can resolve it. The user can still make the allocation irrespectiveof the clahes if he/she wants to. Our work does not build on any existing softwares. It is an effort to ease the workload of our faculties who design our time table every semester.

1.1 Intended Audience

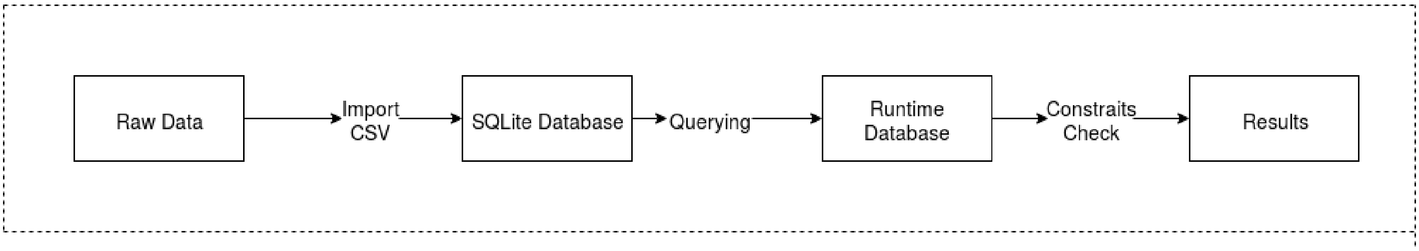
This document is intended for the users including the developers who want to learn and the user who will beusing this tool to make the institute time table.

1.2 References

[1] *Software Requirements Document, Group 9, 15/10/12*

2 Detailed Design

2.1 Architecture



Components

Raw Data (User Dependent)

- These excel/odt files would specify what are the tables and column fields. These need to be converted to CSV format in order to be imported in SQL Database.
- User will be responsible that the CSV files and Database are compatible.

Database

The Database is mainly divided in two parts.

- One with 3 prepopulated tables that are created in the beginning by the user provided raw data. (test.db)
- Other with 2+n tables created at run time in order to practice the constraints and display results, errors and warnings.(test.db)

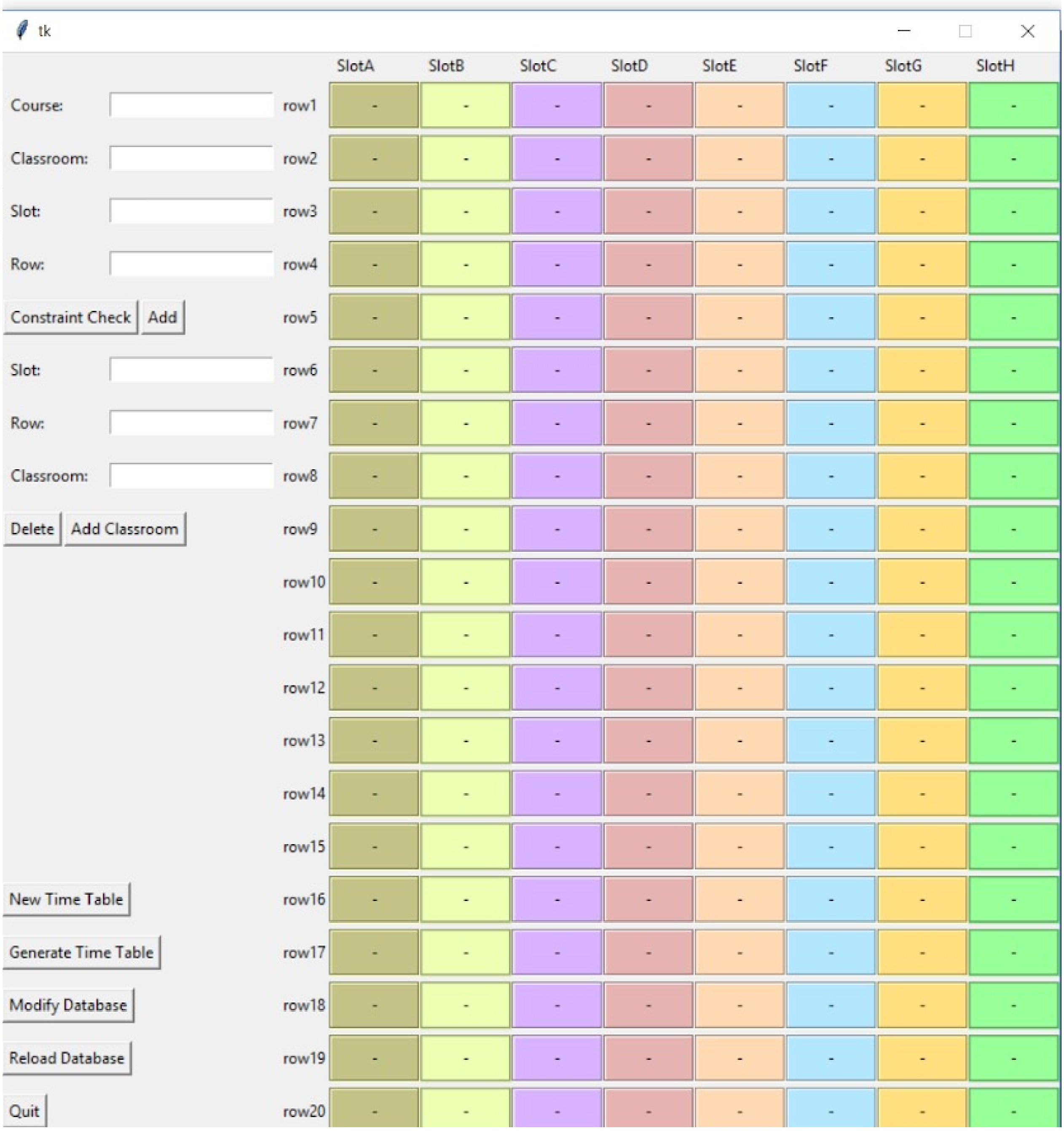
  n is dynamicallly determined from basket.csv at runtime as user can enter 0 or any number of columns in it where each column represents a basket.

Graphic User Interface (GUI)

This is how the user is able to utilize the tool. The interface is created on tkinter module for version 2 of the software. The GUI has variuos input fields wirh the coressponding action buttons following them. The timetable can be visualized as a 8*20 matrix. The same is displayed on the gui and all the changes made by the user are dynamically refelected.

The GUI has been made as interactive, attractive and easy to use as possible, taking cues and user requirements from our mentor (the user himself).

If the user does not wish to get into the deatils of database, or even know about it, he/she can still run the tool easily by following simple instructions from user manual.



	SlotA	SlotB	SlotC	SlotD	SlotE	SlotF	SlotG	SlotH
row1	-	-	-	-	-	-	-	-
row2	-	-	-	-	-	-	-	-
row3	-	-	-	-	-	-	-	-
row4	-	-	-	-	-	-	-	-
row5	-	-	-	-	-	-	-	-
row6	-	-	-	-	-	-	-	-
row7	-	-	-	-	-	-	-	-
row8	-	-	-	-	-	-	-	-
row9	-	-	-	-	-	-	-	-
row10	-	-	-	-	-	-	-	-
row11	-	-	-	-	-	-	-	-
row12	-	-	-	-	-	-	-	-
row13	-	-	-	-	-	-	-	-
row14	-	-	-	-	-	-	-	-
row15	-	-	-	-	-	-	-	-
row16	-	-	-	-	-	-	-	-
row17	-	-	-	-	-	-	-	-
row18	-	-	-	-	-	-	-	-
row19	-	-	-	-	-	-	-	-
row20	-	-	-	-	-	-	-	-

Pre Requisites

- *Python 3 (mostly preinstalled, preferably version 3.6 and above)*
- Tkinter Library (sudo apt-get install python3-tk or sudo apt-get install python3.6-tk)
- SQLite3 (sudo apt-get install SQLite3)

2.2 External Data

Databases

The database has been designed using as less number of tables as possible. No unnecesary columns are present. The three main tables are:

1. Course_Data
2. Professor_Data
3. Classroom_Data

The remaning tables are:

4. Allocated_Subjs
5. Instructore_Slots
6. Basket_n

n = 1 to N (total number of baskets)

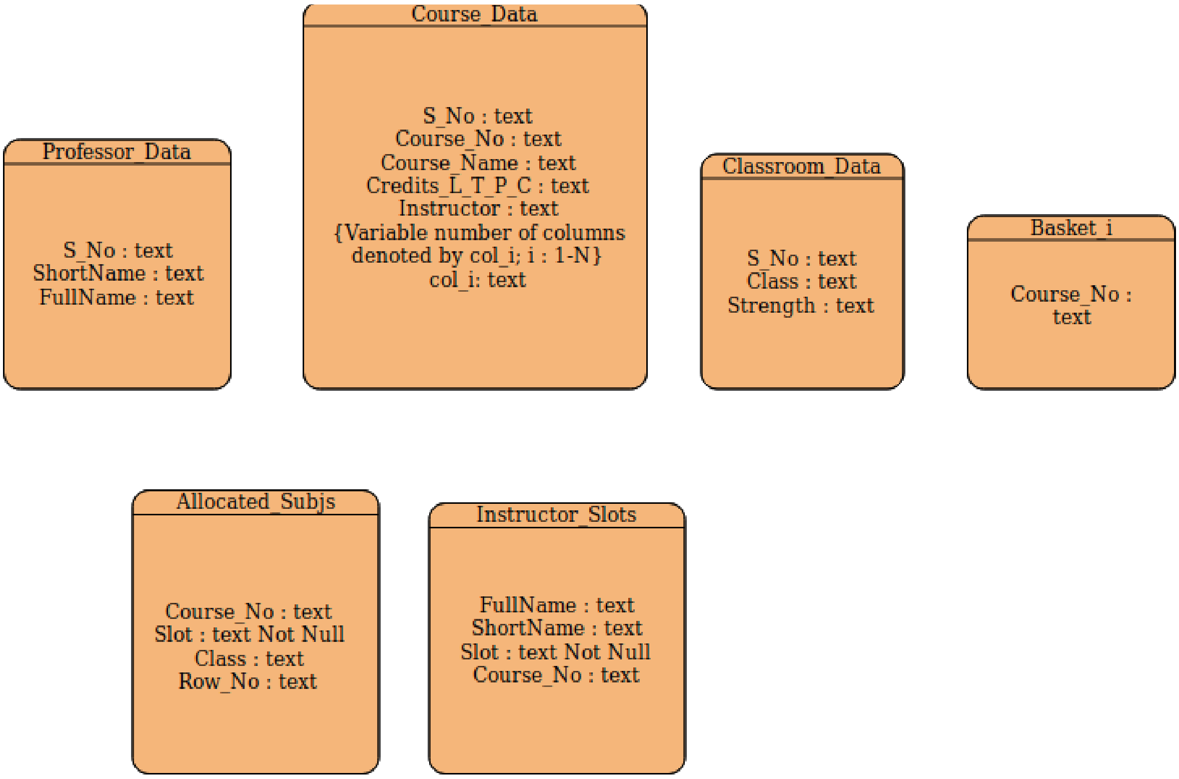
Tables 1,2 and 3 have been created by reading the coressponding .csv files. If the user wishes to modify these tables:

1. Make changes to the .csv files

- Enter correct data and follow the same format
- Save changes and Reload Database using GUI button

2. Make changes only to the tables

- Use GUI Button Modify Database



Files

class.csv, courses.csv, teachers.csv, basket.csv, test.db

1. courses.csv -> Course_Data

S.no.	Course No	Course Title	Credits L-T-P-C	Instructor	Something	--

2. teachers.csv -> Professor_Data

S.no.	Short Name	Full Name

3. class.csv -> Classroom_Datat

S.no.	Class Name	Class Strength

4. basket.csv -> Basket_n

Basket 1	Basket 2	---

2.3 Performance

The product is targetted towards use by a single user at a time as was discussed during the meeting with our mentor and the targetted user. It doesn’t need a server or network. As such the performance is optimal.

2.4 Test Scripts

The following scripts were used. All scripts are written in Python, unless otherwise noted.

tool.py: populate the database for a slot specified by the user.

test.py: create databse from .csv files

allocate.py: create the runtime tables

gui.py: A test and GUI script.