*Grenze International Journal of Engineering and Technology, Jan Issue*

# Analysis and GUI Implementation of Stable Marriage Algorithm for Student Project Allocation

Rose D Pynadath[1] and Sateesh Kumar Awasthi[2]
[1-2]Dr BR Ambedkar National Institute of Technology Jalandhar, Department of ECE, Punjab, India
Email: rosedennypynadath@gmail.com, awasthisk@nitj.ac.in

***Abstract*— Aiming at the problem of number of students and available projects with the teachers, the student-project allocation (SPA) problem emerges. In this paper, we address SPA problem by using stable marriage algorithm (SMA) based on Gale Shapley algorithm. We consider list of choices from students and teachers, which are our participants in SPA. The SMA then uses the preference list to find stable matches among the participants. The stable matches will be found when the algorithm terminates. Furthermore, we use graphical user interface (GUI) connected to SQLite database to maintain the preference list of participants. In the subsequent stages of implementation, we use the database maintained through the GUI. The online storage of preference list avoids data loss and make it easy to use when number of participants increases. It is a sophisticated way to access and modify database when required. Combining the SMA and GUI we address SPA challenges and it can also be used to solve problems in relatable contexts where stable matches are required to be allocated.**

***Index Terms*— GUI, SQLite, Stable Matches and Student Project Allocation.**

## I. INTRODUCTION

In most of the academic institutions, it is required from students to work on a project in the field of their specialization, it is part of student's course completion criteria. The students are provided with projects presented by the supervisors. Each supervisor will be having two or more projects he/she is working on and required assistance. Usually, there are more projects than the students [1]. Students are then asked to submit a preference list containing projects ranked from best to worst based on their interest and skill set. Each teacher will also have choice of students they prefer to supervise, by analyzing their interests, skills, experience in the respective fields of specialization. While a teacher typically has choice regarding the students that he or she is willing to supervise, each student has a list of choice over the projects they prefer to work. Both set of participants have their preference list. When more than one student prefers the same project, it give rise to the problem of student project allocation (SPA) problem. To address this issue, we are using stable marriage algorithm (SMA) to find the most stable match of student and project which can be grouped together [2]. Stable marriage algorithm, which is also known as the Gale-Shapley algorithm, is an efficient and accurate algorithm to find stable matches. Each participant would list their order of preference from best to worst. The algorithm iterates several times to find the stable matches, till both sets of participants are satisfied [3]. The stable matching theory finds its application in various fields including economics, computer science and mathematics to find stable matches. Different methods existing in matching theory are one-sided matching, two-sided matching and many-sided matching. In two-sided matching the variants in computation are one to many, many to one, and many to many matching [4]. Our work is based on Gale-Shapley

*Grenze ID: 01.GIJET.10.1.98*

algorithm, where the preference list of both participants is equally important to find the most stable matches. In the end when algorithm terminates, there cannot be a participant without a match. Matching is completely depending on the order of preference submitted by the groups in the case of two sided matching algorithm. SMA can be used in various fields like matching markets. Some of the applications are discussed below [5]

- *Quarter Allocation* : To allocate quarters to residents, preference list of both are collected. Used in the matching algorithm to find stable matches.
- *College Hostel*: The stable matching method can be used to pair up roommates in hostel, depending on the preference list created by the students considering their class schedule and routines.

Another challenge we are considering in this work is managing the preference list of participants. When the size of participants is in hundreds conventional method to store and collect database fails. In this case, when we apply it at the university level, the number of participants are high, hence the large data size of preference list make it difficult to store it in offline manner [5]. To address this issue we are implementing a system in which preference list of participants are entered into the GUI connected to SQLite database. With this we can store and display the preference list and other information of the participants in the online platform we are creating. There is also a provision to modify or delete the data as per the requirements. Thus, data loss can be prevented and increase the efficiency of the system by minimizing human errors [6]. In this study we find stable matches among students and projects, then store the preference list of both participants in a database through GUI. Thus making the application efficient and user friendly.

The remainder of the paper is structured as follows. The synopsis of related work is covered in Section II. In Section III, the suggested incentive model is presented. We have covered the building blocks of the workflow diagram, such as the algorithm and GUI implementation, in this section. The result and analysis of the system that was put into place are covered in Section IV. And finally, Section V, concludes the paper.

## II. Related Works

In this section, different techniques for finding stable matches in various applications are discussed. These methods have been studied and analyzed from the works of [6], [7], [8], [9], [10], [11], [12]. The authors of [6], [7], [8] have presented an algorithm which is a variation of two-sided matching algorithm. The matching algorithm is using Fuzzy logic to address SPA. The algorithms developed in [6] are student's marks based fuzzy method and incomplete preference list of projects based fuzzy logic. Algorithms try to reduce random allocation of projects to students. The authors of [7] and [8] have also suggested a fuzzy logic approach to solve SPA problem in university level, considering large number of students and number of projects available.

The incremental algorithm implemented in [9], used manyto-one matching theory, in which set of both participants submit their order of preference. The method is based on a metric score, by taking average of the ranking of preferences of the participants. It is then used for validating the course matched to the student after executing the algorithm.

According to [10] and [11] stable matches is found using Reinforcement learning (RL) method. With insight of Markov decision processes, [10] has implemented a framework for teacher-student learning. The algorithm acts as the teacher, advising the student's neural network to take actions predicted by the DRL. In the work of [11], a method which utilizes allocation of resources by taking into account real-time feedback is discussed. RL needs lot of data which can exhaust the system and thus reduce the quality of result. The research work in [12], is based on a model to allocate satellites to unused ground network spectrum.

In this section we have discussed about the methods and approaches suggested in different papers about maintaining and connecting to database. The work in [13], discussed the issue of self generation of android database. They have suggested a methodology for syncing SQLite database and application programming interfaces (API). A technique popular in android technologies is used to create the suggested method in [14] is used to automate the creation of database and API. The tool they developed is driven by model engineering technology. It make use of XML language, popular in computer science. The suggested model is simulated in ARM-Linux environment. It has structure and behaviour of API. They created a simulation setup to test the SQLite data base. The GUI platform created is based on embedded system. The data managed using the embedded system, is rooted in guiding theory.

In the GUI created in the work of [15] is based on Database Management System (DBMS) tool .NET. It is convenient for dealing with database. They created GUI, which do not require querying to access the database connected to it using code like RUD in DBMS. The RUD command is used to perform read, update, write operation.

In their research [16], [17], [18], they have explored different application based on SQLite. Application ranging from internet of things to embedded system. The authors of [16] developed a GUI-based SQLite database system

for Internet of Things (IoT). The model enabled the users to instantly store and retrieve sensor data. It is used in the applications which involves sensor and warnings. The data collected from sensor can be stored and utilized using SQLite database. In [17], [18], authors have worked on embedded system to create database for variety of applications ranging in size and functioning. These studies demonstrate that utilizing a GUI and SQLite to build database applications is generally advantageous and effective. The authors of [18], used the GUI connected to SQLite to mainly store the statistical information of students admitted in degree colleges. Along with SPA problem we have addressed the issue of database management for the preference list of the participants in SMA.

## III. Work Flow Diagram

In order to implement the SMA, for solving SPA problem, we used four stages as shown in Fig. 1. Getting input data in the form of preference list from students and teachers is the first step in the model. After collecting the input data, it is then passed to next stage of implementation which is stored in SQLite and also used in SMA to find stable matches. The code written for GUI implementation in python is connected to SQLite database using the connect() method. The list of choices of students and teachers are used as input data in SMA which produce stable matches of student and project. Given database is used both in GUI and stable matching [21]. The different stages of implementation are as follows:

- Student Preference List
- Teacher Preference List
- Input Data
- SQLite
- Stable Matching Algorithm
- Graphical User Interface
- Student Project Allocation

### *A. Student Preference List*

It is important to collect the preference list to ensure each student get the projects from their order of preference [18]. To collect preference list to find stable matches by executing algorithm, we obtain them in the following form. Q is set of students, $q_i$ represent each student in the set of Q, where $q_i \epsilon$ Q. Q = { $q_1$, $q_2$, $q_3$... }. Table I shows the list of choices of students to the available project *wj* from the teachers $t_k$. In this example we have considered six students and six projects. The preference list of each participants are given in the following tables.

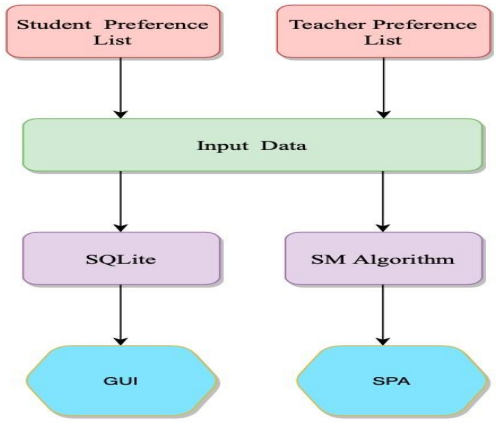


Figure 1. Diagram showing the suggested model's workflow

### *B. Teacher Preference List*

Similarly, we have to obtain the teacher's list of choices in the following form. T is set of teachers, *tj* is element in set T. Below table shows the number of teachers we are considering and projects with each teachers.

Here, we have considered 3 teachers, with total of six projects. TABLE II indicates list of projects with each teachers. TABLE III represent the preference list submitted by each teachers for the projects they are have with them.

These projects are allocated to students, after considering the order of preference of both teachers and students, which are the participants of SMA. The number of projects taken and available would be updated through out the iteration of SMA.

*C. Input Data*

Input data comprises of preference list of students and teachers. After acquiring the list of choices of both sets, we completed collecting the data required as input for implementing GUI and input to the algorithm to find stable matches [15]. Therefore we can proceed to the subsequent stages, as explained below.

*D. SQLite*

Most of the applications require a database to work efficiently. The advantage of implementing GUI with Python is, SQLite is an inbuilt feature of it. Hence by making use of Python libraries we can connect the GUI to SQLite database and use the code to perform different entries and modification in the database. The GUI is connected to SQLite database through connect() method. The keyword calls cursor in the code to query the database. The preference list is stored in the database in format of table. The SQLite command connector .execute() method is used to interact by connecting with the database. The preference list of the participants is entered through GUI and stored in SQLite database.

**Algorithm 1**: Student Project Allocation

```
Assign each student to be available;
while current student q is available do
    begin matching;
    w = first project on q's preference list ;
    if w is available then
    assign q and w to be paired to form a stable
    match ;
    else
        if w prefers q to current student q' then
            assign q and w to be matched and q' to
            be made available
        else
            w removes q and w to be matched and
            q' to be
        end
    end
end
Result: output is the stable matches of n pairs of
students and projects
```

*E. Stable Marriage Algorithm*

The SMA is summarized in the given algorithm above. Each participants are initially available and waiting to be matched. The preference list is collected from participants of both sets. As in the Gale-Shapley algorithm, one of the set is chosen as the "proposer" to make proposals. Choose project or student as the "proposer", both would find stable matches, but matches would be unique. In our work we have considered students as the "proposer" group and teacher as "acceptor" [1]. The group of students q, provide a list of preferred projects and teacher t provides information on the availability of projects w. The algorithm is executed and in each iteration students request to most preferred projects in the order of preference list. The project can tentatively accept the assignment and break it in future if student of higher order preference, request for assignment. The algorithm also addressed the issue when multiple students prefer same project. SMA resolved it by assigning students to project from their next favourite in the order of preference given as input. The participants are unable to initiate request to the ones which have rejected before. But SMA allows to break tentative assignments and proceed to next round of iteration to find potential matches. Number of iterations in SMA is $n^2$, where n is the number of elements in each group [19].

*F. Graphical User Interface*

The GUI is implemented in Python and connected to SQLite database using SQLite commands. Using the Python libraries such as Tkinter, SQLite, Tkt, datetime, messagebox, and tkcalendar we can set the font, colour, size, data entry format of GUI. The GUI frame for storing the preference list of the participants is shown in Fig 2. Inside the frame on left side, we can see the input buttons to insert data of participants like name, contact, branch, email, preference list. Using the method "sqlite3.connect('Preferencelist.db')", SQLite database file with .db extension to store the information of the participants is created [15]. The data we entered in GUI is stored in a table format in the database. We created the table named "Preferences" in the database by using the keyword "CREATE

TABLE". To query the database we used the "conn.cursor()" method and SQLite statements such as SELECT, INSERT, UPDATE, DELETE in the code to enter or modify the data. By using Python library like tkinter.messagebox, GUI can also send interactive messages to user while entering data. "Students Preference List" and "Teachers Preference List" are stored in SQLite database which is connected to GUI. After the execution of GUI implementation code, a file with .db extension which contain the information of participants is created. SQLite commands in the code is used to perform different tasks on the input data of participants, stored in the database [20].

TABLE I. STUDENT'S PREFERENCE LIST

| | |
|---|---|
| $q_1$ | $w_4, w_1, w_5, w_3, w_2, w_6$ |
| $q_2$ | $w_1, w_6, w_2, w_5, w_3, w_4$ |
| $q_3$ | $w_5, w_1, w_2, w_3, w_4, w_6$ |
| $q_4$ | $w_4, w_5, w_3, w_1, w_2, w_6$ |
| $q_5$ | $w_4, w_2, w_1, w_5, w_6, w_3$ |
| $q_6$ | $w_6, w_1, w_2, w_4, w_3, w_5$ |

TABLE II. LIST OF PROJECTS WITH TEACHER

| | |
|---|---|
| $t_1$ | $w_1, w_2$ |
| $t_2$ | $w_3, w_4, w_5$ |
| $t_3$ | $w_6$ |

TABLE III. PREFERENCE LIST OF PROJECTS

| | |
|---|---|
| $w_1$ | $q_1, q_2, q_3, q_4, q_5, q_6$ |
| $w_2$ | $q_3, q_4, q_5, q_6, q_1, q_2$ |
| $w_3$ | $q_4, q_3, q_1, q_6, q_5, q_2$ |
| $w_4$ | $q_5, q_1, q_2, q_4, q_3, q_6$ |
| $w_5$ | $q_1, q_4, q_2, q_3, q_6, q_5$ |
| $w_6$ | $q_2, q_1, q_3, q_6, q_4, q_5$ |

### *F. Graphical User Interface*

The GUI is implemented in Python and connected to SQLite database using SQLite commands. Using the Python libraries such as Tkinter, SQLite, Tkt, datetime, messagebox, and tkcalendar we can set the font, colour, size, data entry format of GUI. The GUI frame for storing the preference list of the participants is shown in Fig 2. Inside the frame on left side, we can see the input buttons to insert data of participants like name, contact, branch, email, preference list. Using the method "sqlite3.connect('Preferencelist.db')", SQLite database file with .db extension to store the information of the participants is created [15]. The data we entered in GUI is stored in a table format in the database. We created the table named "Preferences" in the database by using the keyword "CREATE TABLE". To query the database we used the "conn.cursor()" method and SQLite statements such as SELECT, INSERT, UPDATE, DELETE in the code to enter or modify the data. By using Python library like tkinter.messagebox, GUI can also send interactive messages to user while entering data. "Students Preference List" and "Teachers Preference List" are stored in SQLite database which is connected to GUI. After the execution of GUI implementation code, a file with .db extension which contain the information of participants is created. SQLite commands in the code is used to perform different tasks on the input data of participants, stored in the database [20].

### *G. Student Project Allocation*

The objective of the algorithm is to solve SPA problem by considering the preferences of both sets of participants. SPA is the last step in workflow diagram for the model shown in Figure 1. In this work we have used SMA to solve SPA problem. The students are allocated to projects based on the preference list given in the SMA algorithm given above. Based on the ranking of both group's preferences, student project allocation is done [21]. After allocation, there would be no elements of both the sets left out without matches, as we assumed an equal-sized group. When the algorithm terminates we have the stable matches of students and projects as shown in Fig. 4.

## III. RESULTS AND ANALYSIS

Multiple approaches have been applied to address student project allocation problem (SPA). We have developed a stable marriage algorithm (SMA) based on Gale Shapely to address the SPA problem by finding stable matches. As the number of participants in the SMA increases, it become difficult to manage the database. To overcome this challenge, we used the GUI connected to SQLite database, which enhanced the efficiency of the model by storing preference list in online database. The Table I shows preference list of students. We have considered 6 students as example in this work. Q denotes set of students and individual elements are denoted by q, projects are denoted by

w. Set of teachers denoted by T. Individual teachers are denoted by t as given in Table II. The preference list of students for projects is depicted in Table III. The preference list given in the tables are used as input data to the algorithm to find the stable matches. The GUI implemented is shown in Fig. 2. The algorithm and GUI can be used to solve problems from similar contexts. The database of 6 students and teachers entered into GUI is given in Fig. 3. In Fig. 4, we can see the final stable matches after the algorithm terminated. In Fig. 5 and Fig. 6 we can see the simulation with sets of varied size 16. Fig. 5 shows the GUI after adding preference list of 16 students and teachers. In Fig 6., the SPA simulation with size of 16 is given. The algorithm we implemented in Python solved SPA problem and GUI connected to SQLite database helped to store preference list of the participants. It can be used in applications in similar fields.

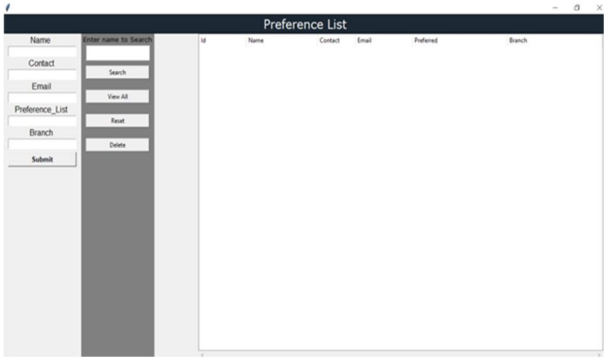


Figure 2: GUI of Preference List

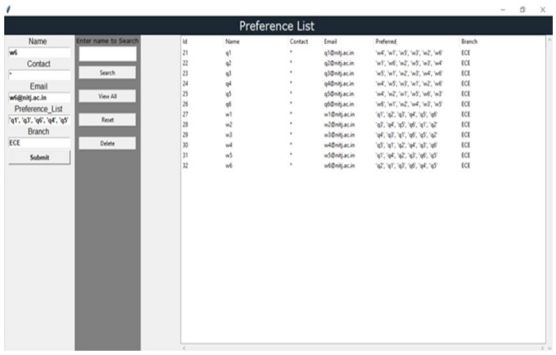


Figure 3: GUI of Preference List after inserting input data

```
Student Project Allocation:
q1 is matched with w4
q2 is matched with w1
q3 is matched with w2
q4 is matched with w3
q5 is matched with w6
q6 is matched with w5
[['q1', 'w4'], ['q2', 'w1'], ['q4', 'w3'], ['q3', 'w2'], ['q6', 'w5'], ['q5', 'w6']]
q1 w4
q2 w1
q4 w3
q3 w2
q6 w5
q5 w6
```

Figure 4: SPA from the results of SMA

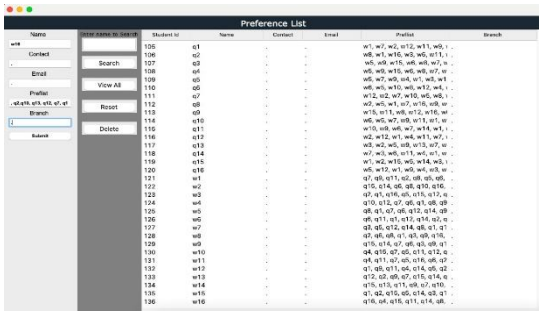


Figure 5: GUI of Preference List with size 16 × 16

```
Student Project Allocation:
q1 is matched with w12
q2 is matched with w8
q3 is matched with w9
q4 is matched with w4
q5 is matched with w7
q6 is matched with w6
q7 is matched with w10
q8 is matched with w5
q9 is matched with w11
q10 is matched with w16
q11 is matched with w14
q12 is matched with w15
q13 is matched with w13
q14 is matched with w1
q15 is matched with w2
q16 is matched with w3
[['q14', 'w1'], ['q8', 'w5'], ['q5', 'w7'], ['q1',
'w12'], ['q12', 'w15'], ['q7', 'w10'], ['q16', 'w3'],
['q15', 'w2'], ['q2', 'w8'], ['q6', 'w6'], ['q3', 'w9'],
['q9', 'w11'], ['q4', 'w4'], ['q13', 'w13'], ['q11',
'w14'], ['q10', 'w16']]
```

Figure 6: SPA with size 16 × 16

## III. CONCLUSION

In this work, we have developed and analyzed stable marriage algorithm (SMA) in Python, with GUI connected to SQLite database. The student project allocation (SPA) problem and data storage issue is resolved using the model. We considered set of students and teachers as our participants in the SPA problem. The preference list of participants are stored in database through GUI. After executing SMA using the preference list of the participants, stable matches of students and projects are found. The allocations found are most preferred by the participants. The algorithm was tested by varying the size of participants. It was able to find stable matches from the given size

of sets. The data inserted using GUI can store, display, modify or delete according to the requirement. Since real-time data monitoring is challenging, we employed a predetermined list of choices for testing and creating SMA. In the future, we can increase the efficiency of the system by adding a feature which automatically notifies the participants when the SMA completes it's execution.

## REFERENCES


[1] Gale, D.; Shapley, L. S. (1962). "College Admissions and the Stability of Marriage"

[2] Sateesh Kumar Awasthi, Yatindra Nath Singh, Simplified Biased Contribution Index (SBCI): A mechanism to make P2P network fair and efficient for resource sharing, Journal of Parallel and Distributed Computing, Volume 124, 2019.

[3] A. H. A. El-Atta and M. I. Moussa, "Student Project Allocation with Preference Lists over (Student, Project) Pairs," Second International Conference on Computer and Electrical Engineering, Dubai, United Arab Emirates, 2009, pp. 375-379.

[4] David J. Abraham, Robert W. Irving, David F. Manlove, Two algorithms for the Student-Project Allocation problem, Journal of Discrete Algorithms, Volume 5, Issue 1, 2007, Pages 73-90, ISSN 1570-8667

[5] V. Arumugam, P. Singh, K. Padhiyar, R. Manek and S. Sayyad, "Academic Project Information Management System," 2021 Asian Conference on Innovation in Technology (ASIANCON), PUNE, India, 2021, pp. 1-6

[6] Paunovic, Vlatka Tomic, S. Bosnic, Ivana Zagar, M. Fuzzy Approach to Student-Project Allocation (SPA) Problem (2019).IEEE Access. PP. 1-1.10.1109/ACCESS.2019.2941730.

[7] D. Srinivasan and L. Rachmawati, "Efficient Fuzzy Evolutionary Algorithm-Based Approach for Solving the Student Project Allocation Problem," in IEEE Transactions on Education, Volume. 51, no. 4, pp. 439-447, Nov. 2008

[8] L. Rachmawati and D. Srinivasan, "A hybrid fuzzy evolutionary algorithm for a multi-objective resource allocation problem," Fifth International Conference on Hybrid Intelligent Systems (HIS'05), Rio de Janeiro, Brazil, 2005, pp. 6

[9] Utture, A., Somani, V., Krishnaa, P., Nasre, M. Student Course Allocation with Constraints. In: Kotsireas, I., Pardalos, P., Parsopoulos, K., Souravlias, D., Tsokas, A. (eds) Analysis of Experimental Algorithms. SEA Lecture Notes in Computer Science(), Volume 11544. 2019 Springer, Cham. https://doi.org/10.1007/978-3-030-34029-24

[10] Z. Liu, W. Lv and X. Ren, "Spectrum Allocation Scheme Based on Stable Matching in Hierarchical Cognitive Satellite Network," in IEEE Access, Volume. 10, pp. 134549-134556, 2022

[11] Y. Zheng et al., "Enabling Robust DRL-Driven Networking Systems via Teacher-Student Learning," in IEEE Journal on Selected Areas in Communications, Volume. 40, no. 1, pp. 376-392, Jan. 2022

[12] X. Ye and T. Fu, "Joint MCS Adaptation and RB Allocation in Cellular Networks Based on Deep Reinforcement Learning With Stable Matching," in IEEE Transactions on Mobile Computing, 2022

[13] Nawahdah, Mamoun Zein, Samer Musleh, Iman Salleh, Norsaremah. (2018). Automatic Generation of Android SQLite Database Components. 10.3233/978-1-61499-900-3-3.

[14] Junyan, Lv Shiguo, Xu Yijie, Li. (2009). Application Research of Embedded Database SQLite. 539 - 543. 10.1109/IFITA.2009.408.

[15] A. T. Raj, G. Nalinipriya, M. Shobana, B. D, G. K. S and H. SP, "A Vibrant GUI Based Data Handling Using Relational Database Framework," 2022 8th International Conference on Smart Structures and Systems (ICSSS), Chennai, India, 2022, pp. 1-6

[16] Yu, Yuanhui. Embedded Internet of Things Applications of SQLite Based on WinCE Mobile Terminal.(2020). 10.21203/rs.3.rs-37411/v1.

[17] L. Junyan, X. Shiguo and L. Yijie, "Application Research of Embedded Database SQLite," International Forum on Information Technology and Applications, Chengdu, China, 2009, pp. 539-543

[18] T. Joyce and C. Hopkins, "Working together: the positive effects of introducing formal teams in a first year Engineering degree", Engineering Education, vol. 6, no. 1, pp. 21-30, 2011.

[19] Dan Gusfield and Robert W. Irving. The stable marriage problem: structure and algorithms. MIT Press, Cambridge, MA, USA.1989.

[20] V. Arumugam, P. Singh, K. Padhiyar, R. Manek and S. Sayyad, "Academic Project Information Management System," 2021 Asian Conference on Innovation in Technology (ASIANCON), PUNE, India, 2021, pp. 1-6

[21] F. -C. Anghel and E. Popescu, "Dynamic Graduation Project Allocation Based on Student-Teacher Profile Compatibility," 2022 International Conference on Advanced Learning Technologies (ICALT), Bucharest, Romania, 2022, pp. 56-60