**SDLC**

**By: Abdallah Salhab**

**\*describe two iterative and two sequential software development lifecycle models:**

\*-the predictive SDLC model:  
in this model you can predict the work flow, the final product, the total cost, the time needed and the scope in the first steps of starting the project.

**\_ The waterfall model:**

To use the waterfall, model the project has to run through the requirement gathering, designing, development, testing, deployment and maintenance phases in sequence and without skipping any phase until it is finished.

It was the first model to be introduced and the easiest to work on.

\_This model has a lot of advantages such as:

-it is easy to understand making it easy for beginner staff to work on it.

- the milestones are easy to understand.

- it is very meticulous.

- it is good to use on very big projects.

\_disadvantages of the waterfall model:

-any problem on the early stages can be very hard to solve.

-the deployment process takes a lot of time to gather the requirements and else.

- the costumer needs to wait for a long time to have an idea on how is the final product going to be.

\*when to use the waterfall mode:

-when the software has been implemented before and the requirements are already known.

-when developing a new version of an existing product.

- when the company wants to make its software work on other platforms.

**\_RAD model:**

RAD or rapid application development model splits the project into components or functions that can be developed at the same time and is all joined up together at the end to finish the project.

The phases of the RAD model:

Business modeling: it takes information that is gathered form the company, this information is used to see how the data can be used to make the industry successful.

Data modeling: information gathered from the business modeling can be used to establish the data objects needed in the business.

Process modelling: in the process modelling stage some changes can be done to the data, during this phase, any descriptions for adding, removing, or changing data objects are also created.

Application generation: it is when all the information is coded, and the first prototype of the system is made, in this phase the data models are turned into prototypes then it can be tested.

\_advantages of the RAD model:

-does not need a lot of time.

-you can have feedback from the customers.

- you can reuse the components.

\_ RAD dis advantages:

* It requires good developers and designers.
* Only the systems that can be separated into models can use RAD.
* The cost of using the RAD model is very high.

\*when to use the RAD model in software development:

-when the time specified to make the software is short.

-when there are a good experienced teams to work on the project.

-when the budget of the project is not important, they have a lot of money.

\*-The adaptive SDLC approach:

It consists of incremental and repeated steps to get to the final product, it is highly adaptive to change in any time.

**\_The spiral model:**

The spiral model is a risk driven model, it uses elements from incremental, waterfall, prototyping models that is why is also called a meta model, and it’s way of doing the development makes it easy to shut down the project at any time with minimal loss.

There are phases that needs to be done in order to complete a loop that ends up with a prototype, if there is a need for more prototypes another loop is needed.

These phases are:  
-planning phase: during this phase the requirements has to be gathered.

-risk analysis: in this phase the risks have to be analyzed and solved if possible.

-engineering phase: in this phase the software has to be developed and tested at the end.

-evaluation phase: in this phase the costumer has to evaluate the end result and see if there is a need for another loop.

\_Spiral model advantages:

-the risks can be avoided at any time.

- additional functionalities can be added later.

-good understanding of the end results from the costumers.

\_Spiral model disadvantages:

-it is a costly model.

-the risk analysis requires highly experienced developers.

-not good for small projects.

- if the risk analysis is bad the project can experience the risk.

\*when to use the spiral model in software development:

-when the requirements are not clear.

- when the project is risky.

-when working on a long term project that is bound to see a lot of significant changes.

**\_The agile model:**

the agile model very similar to the RAD model, but it focuses at building each feature at a time, if the team finishes a feature, they go to the next one.

The agile model releases a part of the final product that is tested every rapid cycle before going to the other feature.

\_agile model advantages:

-customers can see and test every feature after releasing.

-the communication level between customers, developers and managers.

- the software is delivered in a short period of time.

-testing is done for every future.

\_agile model disadvantages:

-there is a lack of designing and documentation.

-the project can be affected by the customer not having a good idea of what he wants in the final project.

-can be a risk for the unexperienced programmers without help.

-it may be hard to assess the amount of effort needed for each future.

\*when to use the agile model in software development:

-when the vision of the project is not well defined.

-when implementing new features in a system.

**\*How to manage the risk in SDLC?**

-In most of the times larger project is far riskier than the smaller ones. Also, if the requirements of the project are not clear problems can happen further more increasing the risk.

-Developing a system that uses commonly used technology is less riskier than if they would do it using new nonstandard technology.

Risk management: risk management is the process in which the risks are identified, evaluate and then eliminating or limiting or avoiding them.

Risk management consists of identifying any risks that can harm the project then prioritize them in terms of the harm they can introduce, after that each risk should have an action plan in case of it happening, the team has to monitor the risk in mind to make sure they do not happen, but if they happen the action plan has to be implemented.

The risk management can save money, allow the team to work faster, make the estimated cost more predictable.

For example:

If there was a game development company that developing a shooter game using the spiral model and right after they released their first prototype, they did a study that showed that a lot of gamer are losing interest in shooting games, after they have identified the risk they studied the impact of the risk if the y continued the development and they figured that it would lead to bankruptcy if they continued in the same way, the organization tried to change the requirements and tried to introduce more features on the game, then they did another study to see if the plan would work, the plan showed that the plan would work but it would not make the success they have anticipated, so it was the stakeholders call to continue or to drop the project and do another one, they chose to drop this project and start another one since that have not invested a lot in the current one yet.

**\* what are benefits of using the Waterfall model in a large software development project?**

The waterfall model is very useful to use in a large project because the requirements has to be determined in the early stage which makes the end goal clear to everyone, it also uses a clear structure when compared to other models which means that it is clearly identifiable in which stage the team is on, further more having a large project means that a lot of the team members might change during the project so when using the waterfall model the new members can cope up more easily than if they would in other models.

**Comparing technical solutions:**

Technical solutions come in hand when the company runs into a risk that is associated with the project size, structure, development team and their familiarity with the system development process.

Two ways to avoid these risks is to either make the project smaller or to hire a new development team that is more experienced with the system development process but each come with a cost.

If the company does not care about the project cost and wants to accomplish the scope as it is the company should hire a new developments team.

But if the company has a limited budget that can’t change, they have to reduce their scope therefor making the project smaller to overcome the risk that is they have a development team with on experience.

**[A] \*Feasibility study:**

The feasibility study is a study made to determine whether or not the project is going to be successful and benefit the organization. This study looks over the economical, technical, organizational, legal, political and the scheduling side of the project.

\*-economic feasibility study (cost-benefit analysis):

This study identifies if the project is worth taking in term of the financial benefits and the costs of the project. It is done in every phase of the SDLC to determine if they should continue on this project or end it or change it.

\*Types of the economical costs:

-Tangible benefits: are the benefit that can be counted; measured in dollars, higher profits.

-Intangible benefits: are the benefits that cannot be counted or measured in dollars, customer satisfaction.

- Tangible costs: costs from the development that can be measured, hardware cost.

- Intangible costs: costs from the development that cannot be measured, inefficiency.

-One-time cost: costs from the project start-up that is done only one time, employee training.

-Recurring cost: the cost associated with running the project, hardware and software maintenance.

\*To determine the project cost which is the cost of the start-up and operating cost and also the procurement and the other related costs and to see if the project can afford them and when the project is going to return it’s costs and how much the organization going to benefit from the project these analysis have to be done like:

-Cash flow analysis: it determines the amount of money available to the company to operate with.

-return of investment (ROI): this function calculates the total amount of money returned from the investment.

ROI = (total benefits – total costs) / total costs.

-break-even point (BEP): the number of years the project needs to return the projects investment.

BEP = number of years with negative cash flow + ((that years net cash flow – that years cumulative cash flow)/that years net cash flow)

\*-technical feasibility study:

It is a study that ensures that the organization has the technical and required resources and technologies to convert their ideas into reality, and if they are not there they would have to source them or develop them.

The technical feasibility study helps the organization avoid risks by making a risk management plan in which the risks has to be identified, analyzes and treated.

The project it-self can change the variables of potential risks for example the project’s size if it is larger it is riskier, developers if they have no experience it is riskier, structure if it is complex it is riskier.

\*-organizational feasibility:

This studies if the project is going to attract customers and how it is going to be affected by and affect the stakeholders, to insure that it is going to be beneficial for them, and if not change the requirements or the project to better suite them.

\*-scheduling feasibility:

It is the organizations estimation on how much time is it going to take to finish the project, it can also give a time goal for each task in the project and it can display milestones.

\*-legal feasibility:

It is to check if the projects break any laws and if it runs with the cyber laws and regulations.

And if any rules was broken the team has to make changes to suit the rules.

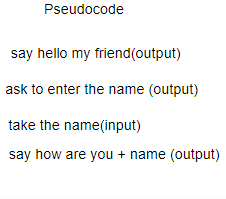
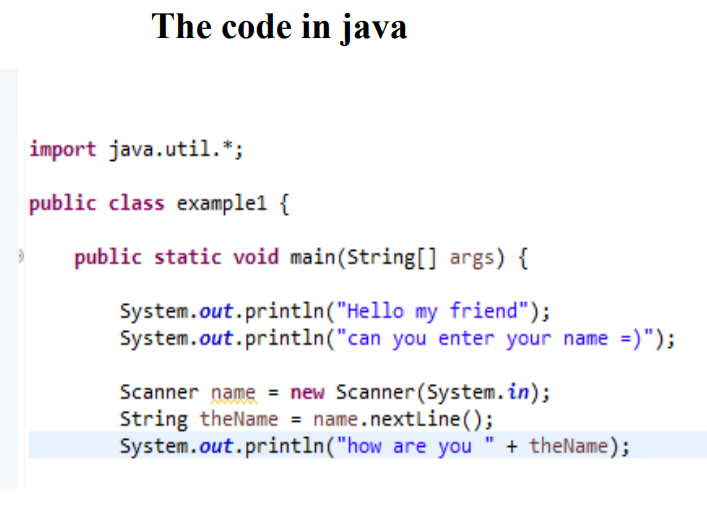
\*-political feasibility:

It sees if the solution to a policy problem going to be accepted from the stakeholders.

If the stake holders did not like the solution, the team has t ogive out other solutions until the stakeholders are satisfied.

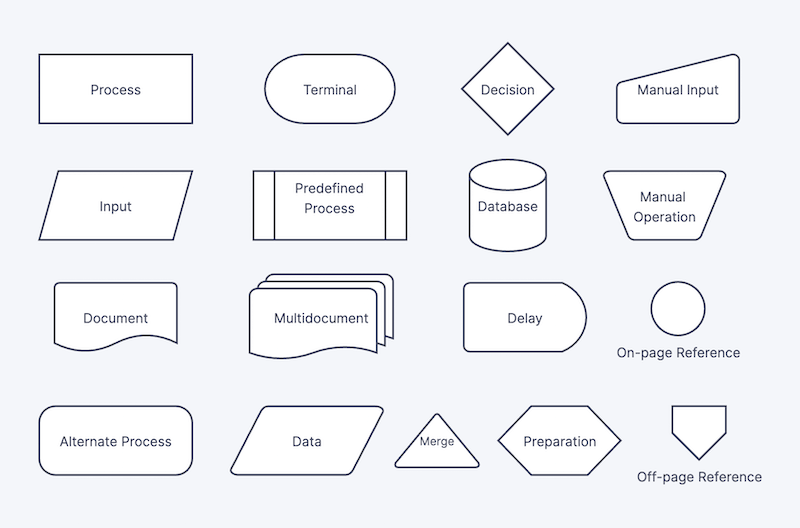
**\*software behavioral design tools and techniques:**

If I have a project in my mind and I need to make a design for it there is a lot of tools that can be used to make it; i.e.



\*Flow chart:

The flow chart gives you shapes that each represent a step, the steps are connected with arrows to show the order of the steps.



-there are two types of flowcharts:

1-high level flowchart: it gives 6-10 steps, these steps are general and not detailed to give a feel on how the project is going to be done.

2- detailed flowchart: it gives those small steps inside the general steps making it more detailed and it is the one that is followed during the project implementation.

We use the flow chart to:

To analyze a process made before.

To standardize a process.

To improve on a process.

To draw a flowchart there is a tool called (Draw.io) that makes it easy to draw because it provides all the symbols needed to draw and end results comes out very tidy.

\*Pseudocode:

It is a way to simply describe an algorithm using common and simple programing language.

It is easier to understand than a code and it is helpful to give a simple description of the algorithm parts.

[C] \*Data flow diagram:

DFD gives a visual representation of the data and how it flows in the system, it gives the input source and the output, and it gives a information about the process that the input data goes through to become outputs.

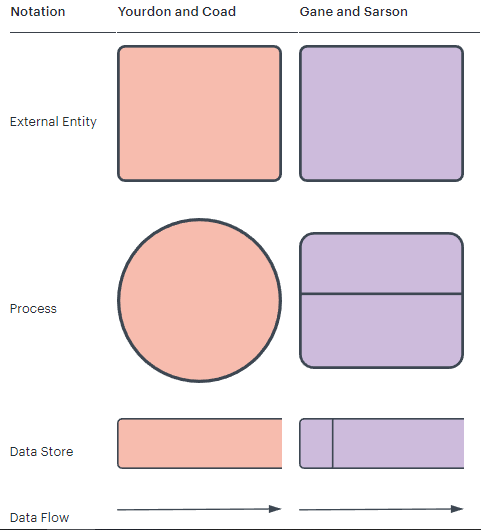
DFD uses symbols to the data and its inputs outputs and the process, and the symbols have three deferent symbols system that accomplices the same thing

\*Gane and Sarson

\*Yourdon and DeMarco

\*Yourdon and Coad

(Yourdon and DeMarco) and (Yourdon and Coad) use the same symbols but (Gane and Sarson) uses slightly different symbols.



DFD can change the level of details in the diagram using levels or layers, the number of levels starts from 0, 1, 2 and some times 3 or higher, and the number of levels depends on the sope and the size of the system.

\*zero level is called a context diagram.

To draw DFDs there is a tool called (smart draw.com) and it allows to draw the diagram in levels and to make sub process inside a process and to link diagrams together.

**[G][B] FSM and EFSM:**

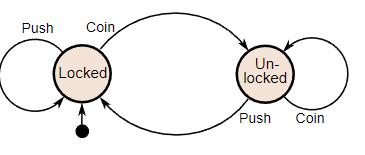
The difference between FSM and EFSM is that a finite state machine transitions between the states with a set of Boolean conditions while in extended finite state machine the transitions happen because of (if statements).

Also, FSM always change in state when giving an input but in EFSM all sets of statements should be inputted before it changes it’s state.

Example of FSM:



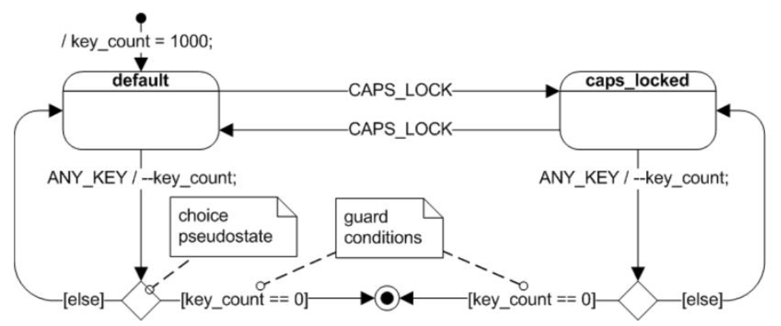
A turnstile: the turnstile has two states that is to be locked or unlocked, the two states can be changed between using two input to put a coin in a slot and to push a button, if the turnstile is locked the arms cant move no mater how many time you are going to try, but as soon as you put a coin in it the state becomes unlocked and then you can push it to pass then it goes back to the locked state, when it is unlocked if you put more coins in it the state wont change and it will remain unlocked.



Example of EFSM:

EFSM can be used to make and run protocols and to test software’s, but a real-life application of it is a keyboard.

Cheap keyboard:



**[F][E] Data driven benefits:**

Data driven software development approach helps organizations improve their software by relying on data to guide the developments process, the data can be collected from deferent sources and one of them is IoT (internet of things), the data driven development involves using KPIs (key performance indicators) to measure the performance of the product then the organization compare the work with OKRs (objectives and key results) to see it they have implemented then and continue the improvement of the software and by that improving the reliability and effectiveness of the software.

Also, data driven development helps the developer with the use of KPIs to monitor the system so the developer would know of any bugs that could have happened because of the inconsistency of the data, so he can solve the problem much faster. Also, data driven can help the developer to know about the client’s desires and behaviors which can help the developer make better dictions while improving the software and by that improving the reliability and effectiveness of the software.

**Software part:**

\*Feasibility study:

\*-economic feasibility:

The idea of the system is not to be a paid service for people to use, but it is a specialized system that helps a certain organization (mostly a school) to make a process more efficient by reducing the time and resources needed to operate it, so making money is not the purpose of the system.

-tangible benefits: the organization won’t have to buy pencils as much, files, storage cabinets any more about 100 JD a year.

-intangible benefits: it is easier for the workers to work with and it is more time efficient and the data can be easily bucked up.

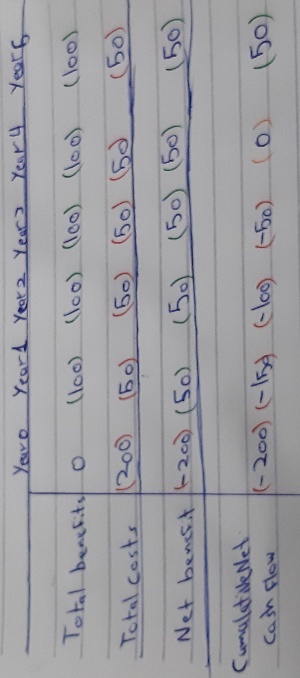
-tangible costs: the cost of building the software that is bout 250JD, and the price of a computer and a printer that is 120 JD.

-Intangible costs: the project acceded the time line by two days.

-One-time cost: installing the system, about 30 JD.

-recurring costs: the cost of the printer’s papers and ink and the price of the maintenance, 50 JD a year.

\_the cash flow analysis:



**\***RIO: RIO = 50 / 450 \* 100% = 11.1%

\*BEP: is 4 years.

\*-technical feasibility:

All the resources and technologies that the company needs to complete the project are available.

In the case of the risk, the project is following the general rules to reduce the risks, like:

-The project size is small.

-The requirements are will defined and the structure is simple

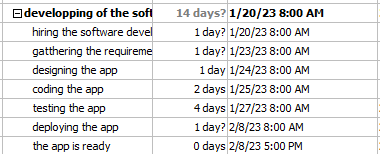
-the one whom are working on the project are experienced.

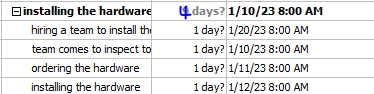
\*-Organizational feasibility:

The stakeholder have agreed on the project and its requirements.

And before the project started a survey was handed out to the staff that worked on the old system to take their impression on the idea and they all had positive feedback.

\*-scheduling feasibility:





\*-legal feasibility:

All the information that is going to be saved on the system are going to be used for the sole purpose of making a better identification for each person on it, and it is not going to be used for something else.

And all the information on the system are going to be general rules the their owners are aware that their info is on this system.

\*-political feasibility:

If there was any problems with any policy, solutions are going to be discussed, and if the stakeholders did not like the solution, they have to give another ones until the stake holders are saticfied.

**\*SRS + SDS:**

**-introduction:**

This code helps schools to tidy up their students, teachers and principal’s information by putting each one in their category, and each one has his own ID.

Their students, teachers and principal’s information have to be pre-entered in the code, and to use it the user has to type in if he wants the information of a student or a teacher or a principal after that he has to inter their ID to get to the information.

**The goals:**

It has to be easy to use.

It has to give the information of the students, teachers and principals.

It has to able to expand the numbers of students, teachers and principals using the code.

**-Functional requirements:**

To store the students, teachers and principles information.

The ability to take their information easily.

The ability to delete or add more their students, teachers and principles information into the system using the code.

**-Nonfunctional requirements:**

The code should have no bugs.

The code should be secure.

The system has to be reliable.

The number of people in the should be able to be scaled up.

The system should be compatible with all schools.

**-Design overview:**  
there are two things that the system can do and that is to store the students, teachers and principals data and the ability to display the information stored for each one of them.

The information storage system:

The students and the teachers and the principals each have their own pre-defined types of information and these can be filled with the persons information. The students, principals and teachers’ numbers can be expanded to fit the needs of the schools.

Displaying the information:

To display the information the system is going to ask if the user is looking for a student or a teacher or a principals information, after the user had answered the system is going to ask for the ID, if the ID is used the information is going to be displayed.

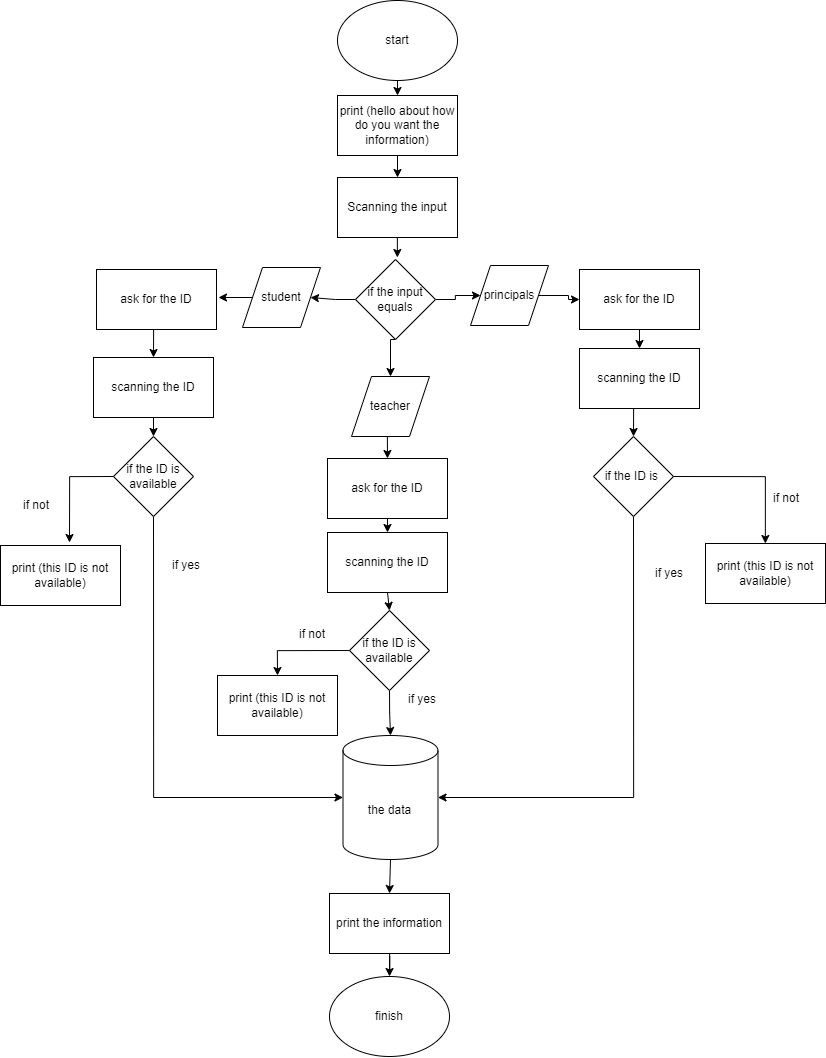
The algorithm:

1-ask for the type of people that you want the information for

2-ask for their ID

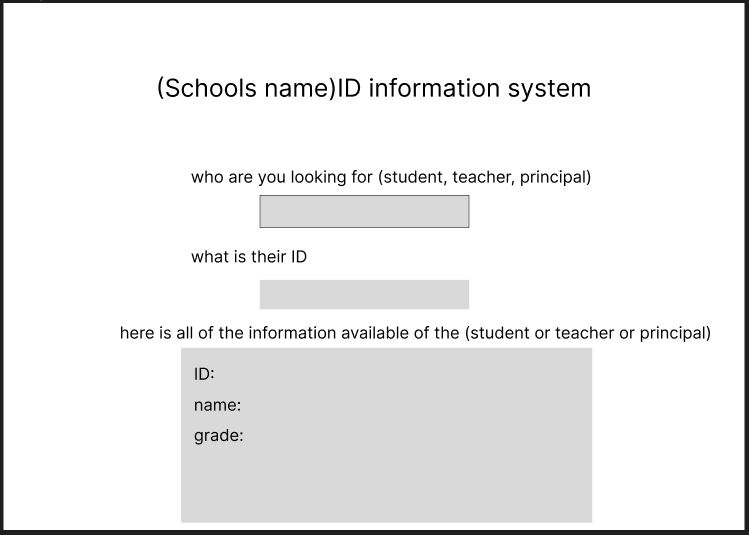
3-the data is going to be pre-defined in the code.

**-Detailed design:**

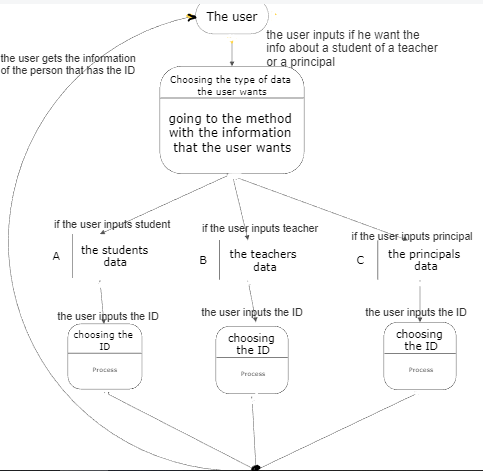


**[D] User interface design:**

It is recommended that the final product is going be presented in this interface.



**Data design:**



**Test plan:**

Firstly, I and going to input all the possible input combinations and check if the data they lead to is the expected data, if one of the combinations did not lead to the expected data I an going to use the debugging feature to check for any logical or syntax bugs.

After that I am going to use Junit testing to test the main method because it is the only one that takes an input and gives a return.

**Conclusion:**

In conclusion there were a couple of sacrifices made because of the lack of experience of the developer like the user interface that is not done and the poor data design.

In the future we would like to make a better user interface, with future of editing and adding data from the interface not from the code.

**RTM:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Rec ID | Rec Description | TestID | Test Description | Test data | Expected result |
| Rec 1 | Changing the information for the students, teachers and principals in the code | TC 1 | Check if the system can add more people | yes | Adding is working successfully |
| TC 2 | Check if the data of the people can be changed | Yes it can be | Data can be changed successfully |
| TC 3 | Check if the system can delete peoples info | Yes | Delete successfully |
| Rec 2 | The user can easily get the data | TC 1 | After asking for the type of person the code scans the input and process accordingly | It gives the right data | It gives the user the expected data |
| Rec 3 | Does it give the user a warning massage if he wrote wrong inputs | TC 1 | After getting asked for the type of people information if it is a student or a teacher or a principal we entered wrong input | It gave the user a warning | It should give the user a warning |
| TC2 | After getting asked for the ID the user typed an invalid ID | It did not give a warning and the code ended | It should give a warning |
| Rec 4 | All the methods and data can be reached | TC1 | I inputted all the possible out comes to see if all the data can be reached | All the data was reached | All the data must be reachable |

[A] <https://in.indeed.com/career-advice/career-development/what-is-technical-feasibility#:~:text=A%20technical%20feasibility%20study%20helps,uncovers%20ways%20to%20overcome%20them>.

[B] <https://en.wikipedia.org/wiki/Extended_finite-state_machine#:~:text=In%20a%20conventional%20finite%20state,a%20set%20of%20trigger%20conditions>.

[C] <https://www.lucidchart.com/pages/data-flow-diagram>

[D] <https://www.figma.com/file/t2PTqHdr6Mloq9iJH2m7Si/Untitled?node-id=0%3A1&t=dGqfFbITkFjNmW1p-0>

[E] <https://www.coursehero.com/file/p7rdn85l/Improving-the-reliability-and-effectiveness-of-the-software-using-data-driven/>

[F] <https://www.cloudzero.com/blog/data-driven-development>

[G] <https://eng.libretexts.org/Under_Construction/Book%3A_Discrete_Structures/09%3A_Finite-State_Automata/9.01%3A_Introduction/9.1.01%3A_Finite-State_Machine_Overview>