**Introduction**

The growing impact over smart phone increases the development of mobile application for educational Guidance and Counseling (GC) .The application is used for Guidance and Counseling services that run on mobile devices and gets updates from a remote web site or desktop app from server. The application is designed specifically for school and college students. The methods are a combination of interactive multimedia approaches and educational psychology. Hence, the design process is carried out with processes of digitizing the material, educational GC services, visualizing wisely and making interactive. The application provides an effective interaction of online counseling for a student for his personal success, drawbacks, problems and feedbacks.

As we all know the students after passing their higher secondary examinations, they need to be counseled in order to choose a stream of their interest for their career. It is not necessary that a student may belong to an educated family and he may need a councilor to get aware about the recent trends. Students who are belonging to the weaker section of society will not be able to get a councilor and here comes a need of a resource that can help the needy students in choosing their righteous choice.

Our idea is to develop a dynamic android application that will guide the students in accordance to their interest. The application will provide a friendly interface for the students of all levels whether graduates or undergraduates .the app will allow the students to choose a field of interest and then the application will show a list of courses that are nowadays available in the various institutes . our android application will be linked with a backend MIS through which a admin can update courses or add new courses, new collages, higher secondary, courses cut off ranges for various courses which on saving on the MIS will directly get displayed to the android app user thus making our android app dynamic in nature .Not only the courses will be shown but a list of institutes with their complete details that are offering the desired course. There will be also complete details available regarding the courses like the course duration, scope, etc.

* The android based application could be handy to provide following information  
  ✔ latest available courses with details  
  ✔ List of available institutions offering the desirable courses  
  ✔ Notification regarding course updates
* The MIS of the project will provide the following features for the admin user.

* Registering institutes , collages, higher secondary’s which are to be viewed on the android app
* Add courses to educational centers like collages, higher secondary’s etc
* Update course duration, cut off admission marks for a particular course.
* Insert or update fee structure details for particular course of an educational centers .

GC Services include areas educational, personal, social, learning and career. GC is a service specifically support / services provided to students so that students can find / understand the self, to know the society, develop themselves in academic results, and plan future.

**Feasibility Study**

The main problems faced by the students after they just finish their tenth standard, they don’t have a proper guideline as it is not necessary that every student would come out from an educated family. They need a best counselor who would guide the students in a right and feasible way according to their choice of interest. Using this app the students may get the idea of future plans and this would help the students to achieve their future goals. The app will be operable on their android smart phones and the students will be updated with the latest scope level of their desired fields or technology. Using this app, students will find it easy to decide the right course and institute for their career.

**Student/User**

r

Management information system for counsel me app

**Student Counselling Application**

**(CounselMe)**

Backend

Database

Fee details

**Courses**

r

**Scholarships**

r

**Institutions**

r

**DFD of Student Counseling Application (CounselMe)**

**Methodology/ Planning of work:**

In [software engineering](https://en.wikipedia.org/wiki/Software_engineering), a software development methodology (also known as a system development methodology, software development life cycle, software development process,  software process) is a splitting of [software development](https://en.wikipedia.org/wiki/Software_development)  work into distinct phases (or stages) containing activities with the intent of better planning and management. It is often considered a subset of the [systems development life cycle](https://en.wikipedia.org/wiki/Systems_development_life_cycle).

In our project we will use the spiral model. It is a risk-driven [process model](https://en.wikipedia.org/wiki/Software_development_process) generator for software projects. Based on the unique risk patterns of a given project, the spiral model guides a team to adopt elements of one or more process models, such as incremental, waterfall, or [evolutionary prototyping](https://en.wikipedia.org/wiki/Software_prototyping#Evolutionary_prototyping).

The Six Invariants

Authentic applications of the spiral model are driven by cycles that always display six characteristics. Boehm illustrates each with an example of a "hazardous spiral look-alike" that violates the invariant.

**Define artifacts concurrently**

Sequentially defining the key artifacts for a project often lowers the possibility of developing a system that meets stakeholder "win conditions" (objectives and constraints).

This invariant excludes “hazardous spiral look-alike” processes that use a sequence of incremental waterfall passes in settings where the underlying assumptions of the waterfall model do not apply. Boehm lists these assumptions as follows:

1. The requirements are known in advance of implementation.
2. The requirements have no unresolved, high-risk implications, such as risks due to cost, schedule, performance, safety, security, user interfaces, organizational impacts, etc.
3. The nature of the requirements will not change very much during development or evolution.
4. The requirements are compatible with all the key system stakeholders’ expectations, including users, customer, developers, maintainers, and investors.
5. The right architecture for implementing the requirements is well understood.
6. There is enough calendar time to proceed sequentially.

In situations where these assumptions do apply, it is a project risk not to specify the requirements and proceed sequentially. The waterfall model thus becomes a risk-driven special case of the spiral model.

**Perform four basic activities in every cycle**

This invariant identifies the four basic activities that must occur in each cycle of the spiral model:

1. Consider the win conditions of all success-critical stakeholders.
2. Identify and evaluate alternative approaches for satisfying the win conditions.
3. Identify and resolve risks that stem from the selected approach(es).
4. Obtain approval from all success-critical stakeholders, plus commitment to pursue the next cycle.

Project cycles that omit or shortchange any of these activities risk wasting effort by pursuing options that are unacceptable to key stakeholders, or are too risky.

Some "hazardous spiral look-alike" processes violate this invariant by excluding key stakeholders from certain sequential phases or cycles. For example, system maintainers and administrators might not be invited to participate in definition and development of the system. As a result, the system is at risk of failing to satisfy their win conditions.

**Risk determines level of effort**

For any project activity (e.g., requirements analysis, design, prototyping, testing), the project team must decide how much effort is enough. In authentic spiral process cycles, these decisions are made by minimizing overall risk.

For example, investing additional time testing a software product often reduces the risk due to the marketplace rejecting a shoddy product. However, additional testing time might increase the risk due to a competitor's early market entry. From a spiral model perspective, testing should be performed until the total risk is minimized, and no further.

"Hazardous spiral look-alikes" that violate this invariant include evolutionary processes that ignore risk due to scalability issues, and incremental processes that invest heavily in a technical architecture that must be redesigned or replaced to accommodate future increments of the product.

**Risk determines degree of detail**

For any project artifact (e.g., requirements specification, design document, test plan), the project team must decide how much detail is enough. In authentic spiral process cycles, these decisions are made by minimizing overall risk.

Considering requirements specification as an example, the project should precisely specify those features where risk is reduced through precise specification (e.g., interfaces between hardware and software, interfaces between prime and sub contractors). Conversely, the project should not precisely specify those features where precise specification increases risk (e.g., graphical screen layouts, behavior of off-the-shelf components).

**Use anchor point milestones**

Boehm's original description of the spiral model did not include any process milestones. In later refinements, he introduces three anchor point milestones that serve as progress indicators and points of commitment. These anchor point milestones can be characterized by key questions.

1. Life Cycle Objectives. Is there a sufficient definition of a technical and management approach to satisfying everyone's win conditions? If the stakeholders agree that the answer is "Yes", then the project has cleared this LCO milestone. Otherwise, the project can be abandoned, or the stakeholders can commit to another cycle to try to get to "Yes."
2. Life Cycle Architecture. Is there a sufficient definition of the preferred approach to satisfying everyone's win conditions, and are all significant risks eliminated or mitigated? If the stakeholders agree that the answer is "Yes", then the project has cleared this LCA milestone. Otherwise, the project can be abandoned, or the stakeholders can commit to another cycle to try to get to "Yes."
3. Initial Operational Capability. Is there sufficient preparation of the software, site, users, operators, and maintainers to satisfy everyone's win conditions by launching the system? If the stakeholders agree that the answer is "Yes", then the project has cleared the IOC milestone and is launched. Otherwise, the project can be abandoned, or the stakeholders can commit to another cycle to try to get to "Yes."

"Hazardous spiral look-alikes" that violate this invariant include evolutionary and incremental processes that commit significant resources to implementing a solution with a poorly defined architecture.

The three anchor point milestones fit easily into the [Rational Unified Process](https://en.wikipedia.org/wiki/Rational_Unified_Process) (RUP), with LCO marking the boundary between RUP's Inception and Elaboration phases, LCA marking the boundary between Elaboration and Construction phases, and IOC marking the boundary between Construction and Transition phases.

**Focus on the system and its life cycle**

This invariant highlights the importance of the overall system and the long-term concerns spanning its entire life cycle. It excludes "hazardous spiral look-alikes" that focus too much on initial development of software code. These processes can result from following published approaches to object-oriented or structure software analysis and design, while neglecting other aspects of the project's process needs.

**User Requirements:**

* Mobile Phone with Android OS 4.0 or Above with GPS enabled, internet connectivity for updating of the app
* Internet Connectivity .
* A desktop pc to run the MIS

**References**

1. *http://www.*tutorialspoint*.com/sdlc/sdlc\_spiral\_model.htm*
2. *Software Engineering by Roger S Pressman*
3. *www.w3school.com*