**Questions: 1. List out 7 functional requirements and 3 non-functional requirements in the new system.**

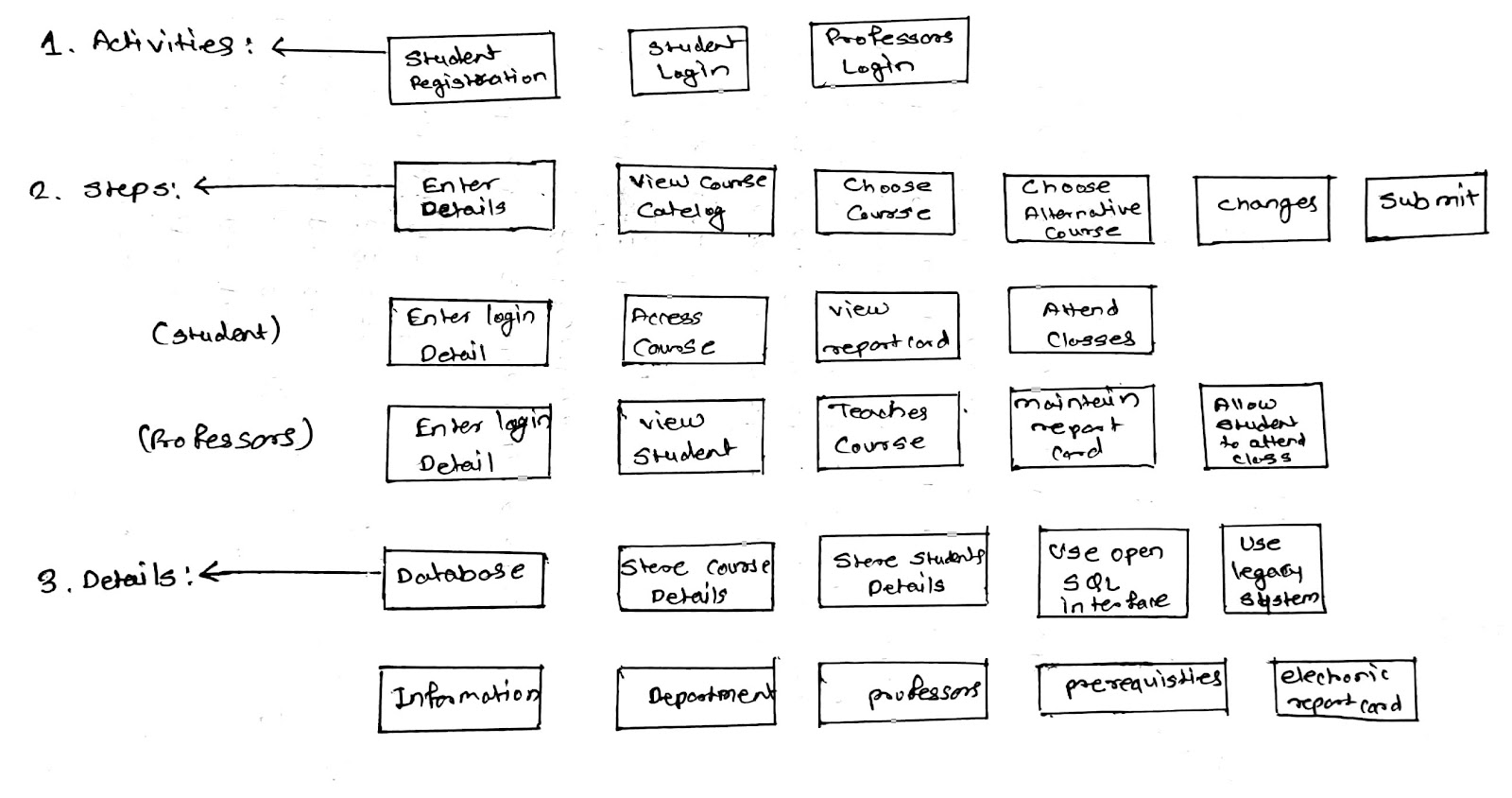
* Functional Requirements:
  + A Registration and Login page
  + Email Confirmation after login
  + A data storage System where all the data of students would be stored
  + Professor can submit grade into database stored
  + Student can view report card with the PC on campus LAN.
  + Student can register for courses and also see information about the courses
  + Registrar close the Registration of Courses
  + Registrar maintain Professor information
  + Registrar main Student Information
* Non-Functional Requirements:
  + The System shall be available 24 hours a day 7 days a week, with not more than 10% down time
  + The system provide access to legacy course catalog database with no more than a 10 second latency.
  + Only Professors can enter grades for students.
  + Only the Registrar is allowed to change any student information
  + The system must prevent students from changing any schedules other than their own, and professors from modifying assigned course offerings for other professors.

**Questions: 2. Write 2 user stories for student’s role and professor’s role. (2point)**

* Student’s role
  + As a Student, I want to register for courses offering from the courses catalog in the current semester
* Professor’s role
  + As a Professor, I want to select the course offering from the courses catalog so that I eligible and wishes to teach in the upcoming semester.

**Questions: 3. Suggest the most suitable software development model to build this system and clarify why you choose this model by the following criteria: (3 points)**

* The various Software development lifecycle options available are as illustrated :
* A project’s quality, timeframes, budget, and ability to meet the stakeholders’ expectations largely depend on the chosen model.
* The types in the lower quadrants of the chart being sequential (development processes - dev, QA, deployment) done in steps rather than iteratively are easy to implement, use and manage. These models imply low customer involvement eg - Waterfall, V-Model
* As you move higher and towards the right side of the quadrant, the process becomes less rigid and offers more flexibility when it comes to changes in the requirements for future software. These models are more cooperative and include customers in different stages of the software development cycle. eg-Scrum, Kanban, XP -collectively called Agile methodology.
* As mentioned in the problem statement - due to federal funding the college cannot replace the entire system at once and thus it is advisable to go about development in phases or cycles. Feedback from the stakeholders - students and teachers are important and requirements understanding is also clear and understood per problem statement.
* A suitable development model would be Agile(scrum). In general, at the heart of Agile are iterative development, intensive communication, and early customer feedback. In Agile we may choose Scrum as a development lifecycle method - Scrum is probably the most popular Agile model. The iterations are usually 2-4 weeks long and they are preceded with thorough planning and previous sprint assessment. No changes are allowed after the sprint activities have been defined leading and have extensive stakeholder feedback and involvement
* The team size of 4 developers, 2 QA and a team lead who usually functions as Scrum master also fits the criteria for the Scrum software development model, as the team can pick development work in phases as required. The requirements for each sprint can be defined based on budget and priority of the functionality, QA in each sprint ensures reliability. There may be a smaller user involvement in the project for sprint planning or feedback and suits this development model.

**Questions: 4.: (3 points)**