

REQUIREMENTS ELICITATION AND SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

DR. JYOTI DESHMUKH

2 TOPICS

- Functional Requirements and Quality Attributes
- Elicitation Techniques Information Description or System Model
- Quality Attribute Workshop(QAW)
- Analysis , Prioritization and Trade off
- Architecture Centric Development Method (ACDM)

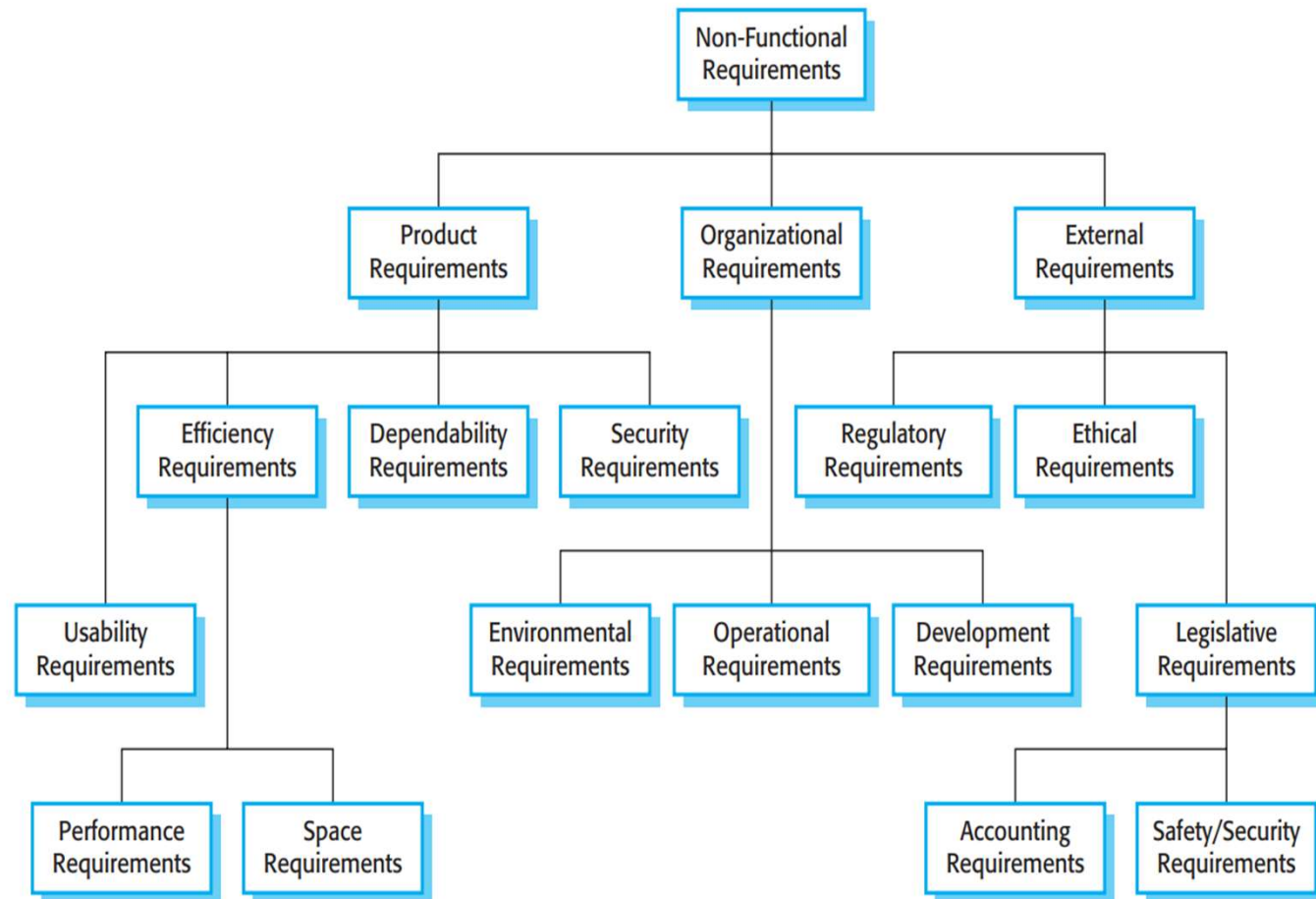
3 TOPICS

- SRS Definition
- Information Description or System Model
- Functional Description
- Requirements Validation
- Ten Tips for Getting Useful Information from Users
- Characteristics of a Software Requirements Specification
- Usable during the operation and maintenance phase
- Rules of Order for Specifying SW Requirements

4 FUNCTIONAL AND NON FUNCTIONAL REQUIREMENT

- Requirements of a system can be classified into functional and non functional (i.e. quality attributes).
- Functional requirements are statements of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situations.
- Quality attributes define global properties of a system. Usually these are only dealt with in the later stages of a software development process, such as design and implementation
- Identify quality attributes. Quality attributes can be assumptions, constraints or goals of stakeholders. By analysing the initial of set requirements, the potential quality attributes are identified.
- For example fundamental quality attribute are Security issue, response time
Other concerns are identified in a similar fashion: Multiuser System, Compatibility, Legal Issues, Correctness and Availability

5 TYPES OF NON-FUNCTIONAL REQUIREMENTS



6 ELICITATION TECHNIQUES

- A major goal of Requirements Elicitation is to avoid the confusions between stakeholders and analysts It is important to distinguish different elicitation methods according to the four methods of communication.
- 1. Conversational
 - Interviews
 - Questionnaire
 - Brainstorming
- 2. Observational
- 3. Analytic
 - •Laddering •Repertory grid
- 4. Synthetic

7 QUALITY ATTRIBUTE WORKSHOP(QAW)

- The Quality Attribute Workshop (QAW) is a facilitated method that engages system stakeholders early in the life cycle to discover the driving quality attributes of a software-intensive system The QAW involves the following steps:
- 1. QAW Presentation and Introductions.
- 2. Business/Mission Presentation.
- 3. Architectural Plan Presentation.
- 4. Identification of Architectural Drivers.
- 5. Scenario Brainstorming.
- 6. Scenario Consolidation.
- 7. Scenario Prioritization.
- 8. Scenario Refinement.

8 ANALYSIS , PRIORITIZATION AND TRADE OFF

-
- The goal of requirement analysis phase is answer to question:
 - what software must do (and with what constraints)?
 - The goal of software analysis phase is answer to question:
 - how system should work?
 - Software engineering elements that are used during analysis phase: notations for model record, methods of model preparation tools for easy use of notations and methods.
 - Prioritizing requirements helps the project team to understand which requirements are most important and most urgent.

9

- Based on this finding a software engineer can decide what to develop/implement in the first release and what on the coming releases.
- Prioritization is also a useful activity for decision making in other phases of software engineering like development testing, and implementation.
- There are a number of techniques available to prioritize the requirements with their associated strengths and limitations.

10 ANALYSIS , PRIORITIZATION AND TRADE OFF

- ATAM(Architecture Tradeoff Analysis Method) STEPS

Step 0 - Planning/Information exchange

Step 1 - Scenario brainstorming

Step 2 - Architecture presentation

Step 3 - Scenario coverage checking

Step 4 - Scenario grouping and prioritization

Step 5 - Map high priority scenarios onto architecture

Step 6 - Perform quality attribute-specific analysis



11 3 P'S

-
- People - stakeholders
 - Product- project
 - Process- models

12 W⁵HH PRINCIPLE

- 1) **WHY** is the system being developed?
Validity of business reasons for software work
- 2) **WHAT** will be done?
Task set which is required for project
- 3) By **WHEN**?
Project schedule
- 4) **WHO** is responsible for a function?
Role and responsibility of each member must be defined
- 5) **WHERE** are they organizationally located?
All roles don't reside with in software team

13

-
- 6) **HOW** will the job be done technically and managerially?

After scope, strategy is need to be build

- 7) **HOW MUCH** of each resource (e.g., people, software, tools, database) will be needed?

Estimates are required



14 SRS DEFINITION

- A set of precisely stated properties or constraints which a software system must satisfy.
- A software requirements document establishes boundaries on the solution space of the problem of developing a useful software system.
- A software requirements document allows a design to be validated - if the constraints and properties specified in the document are satisfied by the software design, then that design is an acceptable solution to the problem.
- The task should not be underestimated, e.g. the requirements document for a ballistic missile defence system (1977) contained over 8000 distinct requirements and support paragraphs and was 2500 pages in length.

15

-
- Six requirements which a software requirements document should satisfy:
 - it should specify only external system behavior,
 - it should specify constraints on the implementation,
 - it should be easy to change,
 - it should serve as a reference tool for system maintainers,
 - it should record forethought about the life cycle of the system, and
 - it should characterize acceptable responses to undesired events.

16 INFORMATION DESCRIPTION OR SYSTEM MODEL

- This conceptual model is a very high-level view of the system in which the major user services are identified and their relationships documented.
- It is necessary to establish an explicit, precisely defined system model at an early stage and to use this model to understand the system.
- The most effective notations for describing the conceptual model of a system are graphical notations - they are usually understandable by users who have no technical background in software engineering.

17 FUNCTIONAL DESCRIPTION

- The functional system requirements are those system services which are expected by the user of the system.
- The analyst must avoid the introduction of implementation concepts in this section.
- In principle, the functional requirements should be both complete and consistent.
- Completeness means that all user-required services are specified.
- Consistency means that no one requirement definition should contradict any other.

18 FUNCTIONAL DESCRIPTION

- There are three ways of expressing the functional requirements of a system:
 - in natural language,
 - in a structured or formatted language which has some rules but no rigorous syntactic or semantic specification, and
 - in a formal specification language with rigorously defined syntax and semantics.

REQUIREMENTS VALIDATION

19

-
- The system requirements should be validated - if they are not, then errors in the requirements definition will be propagated to the system design and implementation and expensive system modifications may be required to correct these errors.
 - There are four separate stages involved in validating requirements:
 - Step 1: The requirements should be shown to be consistent.
 - Any one requirement should not conflict with any other.
 - Step 2: The requirements should be shown to be complete.
 - The definition should include all functions and constraints intended by the system user.

REQUIREMENTS VALIDATION

20

- Step 3: The requirements should be shown to be realistic.
 - There is no point in specifying requirements which are unrealizable using existing hardware and software technology. It may be acceptable to anticipate some hardware developments, but developments in software technology are much less predictable.
- Step 4: The needs of the user should be shown to be valid.
 - A user may think that a system is needed to perform certain functions but further thought and analysis may identify additional or different functions which are required.
- Requirements reviews are the most effective way to validate requirements. During a review, the requirements are studied and considered by both users and software developer.

21 TEN TIPS FOR GETTING USEFUL INFORMATION FROM USERS

- Include real end users, not their representatives.
- Don't ask users to do your job.
- Overcome resistance to change.
- Use data to settle differences of opinion.
- Leave room for users to change their minds.
- Keep an open mind.
- Live in their camp for a while.
- Get some communications help.
- Don't rely on memory or general impressions.
- Don't rush to write things off as too difficult.



22 **CHARACTERISTICS OF A SOFTWARE REQUIREMENTS SPECIFICATION**

- A good SRS is
 - unambiguous,
 - complete,
 - verifiable,
 - consistent,
 - modifiable,
 - traceable, and
 - usable during the operation and maintenance phase.

23 UNAMBIGUOUS

- Every requirement has only one interpretation.
- Each characteristic of the final product is described using a single unique term.
- A glossary should be used when a term used in a particular context could have multiple meanings.

24 COMPLETE

- A complete SRS must possess the following qualities:
 - inclusion of all significant requirements,
 - definition of the responses of the software to all realizable classes of input,
 - conformity to any standard that applies to it,
 - full labeling and referencing of all tables and diagrams and the definition of all terms.

25 VERIFIABLE

- Every requirement must be verifiable.
- There must exist some finite cost-effective process with which a person or machine can check that the software meets the requirement.

26 **CONSISTENT**

-
- No set of individual requirements described in the SRS can be in conflict.
 - Types of likely conflicts:
 - Two or more requirements describe the same real world object in different terms.
 - The specified characteristics of real world objects might conflict.
 - There may be a logical or temporal conflict between two specified actions.

27 **Modifiable**

- The structure and style of the SRS are such that any necessary changes to the requirements can be made easily, completely and consistently.
- Requirements:
 - a coherent and easy-to-use organization (including a table of contents, index and cross-referencing),
 - not be redundant - this can lead to errors.

28 TRACEABLE

- The origin of each requirement must be clear.
- The SRS should facilitate the referencing of each requirement in future development or enhancement documentation.
- Types:
 - Backward traceability
 - Each requirement must explicitly reference its source in previous documents.
 - Forward traceability
 - Each requirement must have a unique name or reference number.

29 USABLE

- Usable during the operation and maintenance phase.
- The SRS must address the needs of the operation and maintenance phase, including the eventual replacement of the software.

30 RULES OF ORDER FOR SPECIFYING SW REQUIREMENTS

- Rule 1: Use an industry standard for
 - a standard format,
 - a completeness check and
 - a checklist of good requirement characteristics to establish consistency between documents.
- Rule 2: Use models for
 - functional relationships,
 - data flow,
 - data structure and
 - performance to express complete requirements.

31 RULES OF ORDER FOR SPECIFYING SW REQUIREMENTS

- Rule 3: Limit the structure of paragraphs to a list of individual sentences
 - to increase the traceability and modifiability of each requirement and to increase the ability to check for completeness.
- Rule 4: Limit the structure of each sentence to a simple sentence (noncompound verbs or objects).
 - This is to increase the verifiability (testability) of each requirement.
- Rule 5: Limit the verbs and objects in the sentences to a small set with a single specified definition for each word.
 - This improves consistency and reduces ambiguity.

32 RULES OF ORDER FOR SPECIFYING SW REQUIREMENTS

- Rule 6: Limit the verbs and the objects to terms that are common to the end user of the product
 - in order to increase user understanding of the requirements.
- Rule 7: Limit the verbs and objects to actions and items that are visible external to the product.
 - This results in a reduction of the amount of design data that goes into the requirements and an increase in the testability of the product.

33 THE SEVEN SINS OF THE SPECIFIER

- Noise
 - The presence in the text of an element that does not carry information relevant to any feature of the problem.
- Silence
 - The existence of a feature of the problem that is not covered by any element of the text.
- Over specification
 - The presence in the text of an element that corresponds not to a feature of the problem but to features of a possible solution. It is typically found in requirements written by programmers. But, implementation decisions taken too early may turn out to be wrong and important problem features can be overlooked.
- Contradiction
 - Two or more elements define a feature of the system in an incompatible way.

34 THE SEVEN SINS OF THE SPECIFIER

- Ambiguity
 - An element that makes it possible to interpret a feature of the problem in at least two different ways.
- Forward reference
 - Implicit forward references (uses of a concept that come before the proper definition of the concept without particular warning to the reader) are the problem. This is why a glossary is so important.
- Wishful thinking
 - An element that defines a feature of the problem in such a way that a candidate solution cannot realistically be validated with respect to this feature.



SRS Document

- 1. Introduction**
 - 1.1 Purpose
 - 1.2 Scope
 - 1.3 Intended Audience
 - 1.4 Definitions, Acronyms and Abbreviations
 - 1.5 References
- 2 Overall Description**
 - 2.1 Product Perspective
 - 2.2 Product Functions
 - 2.3 User Classes and Characteristics
 - 2.4 Operating Environment
 - 2.5 User Environment
 - 2.6 Design/Implementation Constraints
 - 2.7 Assumptions and Dependencies
- 3 External Interface Requirements**
 - 3.1 User Interfaces
 - 3.2 Hardware Interfaces
 - 3.3 Software Interfaces
 - 3.4 Communication Interfaces
- 4 System Features**
 - 4.1 System Feature A
 - 4.1.1 Description
 - 4.1.2 Action/Result
 - 4.1.3 Functional Requirement
- 5 Other Non-Functional Requirements**
 - 5.1 Performance Requirements
 - 5.2 Safety Requirements
 - 5.3 Security Requirements
 - 5.4 Software Quality Attributes
 - 5.5 System Flowchart
 - 5.6 Data Flow Diagram
- 6 Design**
 - 6.1 E-R Diagram
 - 6.2 Database Schema
 - 6.3 User Interface Design
 - 6.3.1 System Block Diagram
 - 6.3.2 Input Output Form
- 7 Other Requirements**
- 8 Glossary**

SRS Document

022

1. Introduction

1.1 Purpose

This specification document describes the capabilities that will be provided by the software application 'Bluetooth Examination (BE)'. It also states various required constraints by which the system will abide. The intended audience for this document are development team, testing team and end users of the product. This project will enable educational institutes to conduct test and have automated checking of answers based on the response by the candidates.

1.2 Scope

This Application will be useful for conducting Examinations like GRE, GMAT, i.e. Multi-Choice Question (MCQ). The Server Application will register students when the student sends the enrolment details via J2ME application on the Bluetooth enabled mobile. The Server application will maintain a database of Questions along with the difficult level. It will generate Random Questions according to the difficulty criteria set for each question. The Question Paper generated will be unlocked by a passcode which will be provided by the faculty. It will generate results once the question paper is submitted by the student.

1.3 Intended Audience

The major audience for this application are schools, colleges, institutes and universities.

1.4 Definitions, Acronyms and Abbreviations

B.E – Bluetooth Examination

1.5 References

- IEEE 830-1998 standard for writing SRS document.
- I. Sommerville, Software Engineering, 8 th ed. England: Addison-Wesley, 2007.

2. Overall Description

2.1 Product Perspective

The product we aim to build will be desktop application as well as mobile application which will be self-contained and independent software product.



(Fig 2.1)

2.2 Product Functions

The System will allow access only to authorized users with specific role like faculty and student. Depending upon the users' role, he/she will be able access only specific module of the system.

- (a) Login Facility for enabling only authorized access to the system.
- (b) Generating Question Paper facility for faculty.
- (c) Faculty will be able to add, modify or delete questions to and from the Questions Database.
- (d) Connect Facility for enrolling and sending question paper to students.
- (e) Passcode Facility for the faculty to lock the question paper which will be broadcasted to the students.
- (f) Students will be able to unlock the question paper on the client application by entering the passcode.
- (g) Facility for the Students to choose the answer and submit to the server.
- (h) Facility for the server application to accept answers from student and to calculate result and send it to respective user.

2.3 User Characteristics

- Educational Level: At least graduate should be comfortable with English language.
- Experience: Should be well informed about the course structure Multi Choice Question and about the theory examination scheme.
- Technical Expertise: Should be comfortable using general purpose application on a computer.

2.4 Operating Environment

As the language used is Java so it will require JRE i.e. Java Runtime Environment.

2.5 User Environment

J2ME supporting Bluetooth enabled device.

2.6 Design/Implementation Constraints

Users at university will have to implement a security policy to safeguard the marks related information from being modified by an authorized user.

3.1 User Interfaces

For Server

3.1.1 Login Screen

It will be first screen that will display on server application. It will allow users to access different screens.

Faculty needs to enter his/her login details like user id, password.

- i. User ID: Alphanumeric of length up to 20 characters.
- ii. Password: Alphanumeric of length minimum 8 characters.

3.1.2 Question Paper Screen

It will allow faculty to add new question, edit them, remove them, and assign weightage to the questions. Faculty has to select which kind of questions he wants to enter whether it is a multi-choice question or a theory question and accordingly insert questions.

3.1.3 Enrolled Screen

It will allow faculty to check which students are enrolled for examination and to whom the question paper are sent. If a student fails to receive the question paper, then faculty can resend the question paper.

3.1.4 Connect Screen

It will allow the faculty to enable the server to accept student's requests for enrolling and receiving the send question paper.

For Client

3.1.5 Warning Screen

It will be a warning message stating that launching any other application will result in exiting from current application.

If any other application is launched, then this application would exit and student won't be allowed to again give the exam.

3.1.6 Enrolment Screen

It will allow students to enrol themselves for appearing in examination. Students would be required to submit their details like Id, name. Also Bluetooth ID will be fetched while enrolling.

10/11/2022

3.1.7 Test Screen

It will display Question Papers on students Bluetooth enabled device once student enters the passcode. As the student enters the passcode, it will start the examination by allowing students to select their respective answer for particular questions. Students can finish the examination by submitting answers.

3.2 Hardware Interfaces

3.2.1 For Server :

- Screen resolution of at least 800 x 600 – required for proper and complete viewing of screens. Higher resolution would not be a problem.
- Standard Bluetooth Dongle for carrying the operations between Bluetooth enabled mobile phone and Server computer.
- Standalone system or network based – not a concern, as it will be possible to run the application on any of these.

3.2.2 For Client :

- Mobile phone with J2ME support.
- Bluetooth Enabled Mobile Phone.

3.3 Software Interfaces

3.3.1 For Server :

- Any windows-based operating system. (Windows XP/Vista/Seven)
- MS-SQL 2008 as the DBMS – for database.
- Java – for coding/developing the software.

3.3.2 For Client :

- Any J2ME-Supporting operating system.
- Sun Java wireless toolkit – for coding/developing the software.

3.4 Communication Interfaces

Bluetooth will be used for communication between the server and the client.

4. System Features

4.1 Passcode

4.1.1 Description

Question papers will be locked or encoded by a passcode using server application. This passcode will be provided to the students by the faculty.

4.1.2 Validity Checks

The passcode entered by the student will be verified for its correctness.

4.1.3 Sequencing Information

The passcode will have to be entered into the client application before the student can start the test.

10/11/2022

4.1.4 Error Handling

If any of the above validations/sequencing does not hold true, appropriate error message will be prompted to the user for doing the needful.

4.2 Question Paper Generation

4.2.1 Description

It provides two options for question paper generation

- Theory Question Paper – This option enables faculty to enter theory question for various topics and assign weightage to them.
- Multi Choice Question Paper – This option enables faculty to enter multi choice questions with four choices for each question, among which one choice is correct.

4.2.2 Validity Checks

In the theory question paper, the question paper generated will be validated whether it has desired total marks.

4.2.3 Sequencing Information

The desired parameter like complexity, marks, etc. will have to be entered before the question paper is generated.

4.2.4 Error Handling

If any of the above validations/sequencing does not hold true, appropriate error message will be prompted to the user for doing the needful.

4.3 Random Questions

4.3.1 Description

Out of various questions entered by the faculty, given number of random questions will be sent to the students. Possibly, different sequence for each student, eliminating the possibility of same question occurring twice.

4.3.2 Validity Checks

The question paper generated will be validated to see whether each question paper abide by above feature.

4.3.3 Sequencing Information

The information about the question scheme should be fed into the server application before the question paper can be generated.

4.3.4 Error Handling

If any of the above validations/sequencing does not hold true, appropriate error message will be prompted to the user for doing the needful.

4.4 Recovery Of Password

4.4.1 Description

Suppose if the faculty forgets his/her password there is a mechanism to recover his password or allow him to set new password.

4.4.2 Validity Checks

When the request for forgotten password is made the system will check the security options validate them .

4.4.3 Sequencing Information

The information about the security functions should be fed into the server to verify the constraints when the request for new password is made

4.4.4 Error Handling

If any of the above validations/sequencing does not hold true, appropriate error message will be prompted to the user for doing the needful.

4.5 Connect Student Manually To The Sever

4.5.1 Description

If any student fails to connect to the server via his device the faculty can manually send another request to him via server and make him connected to the server.

4.5.2 Validity Checks

The software will allow the faculty to check which students has received the paper. In case, if any student has failed to receive the paper then faculty can resend the paper manually using the software.

4.5.3 Sequencing Information

The test paper will be sent from the server to the client via Bluetooth.

4.5.4 Error Handling

If any of the above validations/sequencing does not hold true, appropriate error message will be prompted to the user for doing the needful.

5. Other Non-Functional Requirements

5.1 Performance Requirements

Why None? Give reasons

5.2 Safety Requirements

None.

5.3 Security Requirements

The application will be password protected. Users will have to enter their correct username and password in order to access the application.

5.3.1 Software Quality Attributes Maintainability

The application will be designed in a maintainable manner. It will be easy to incorporate new requirements in the individual modules

5.3.2 Portability

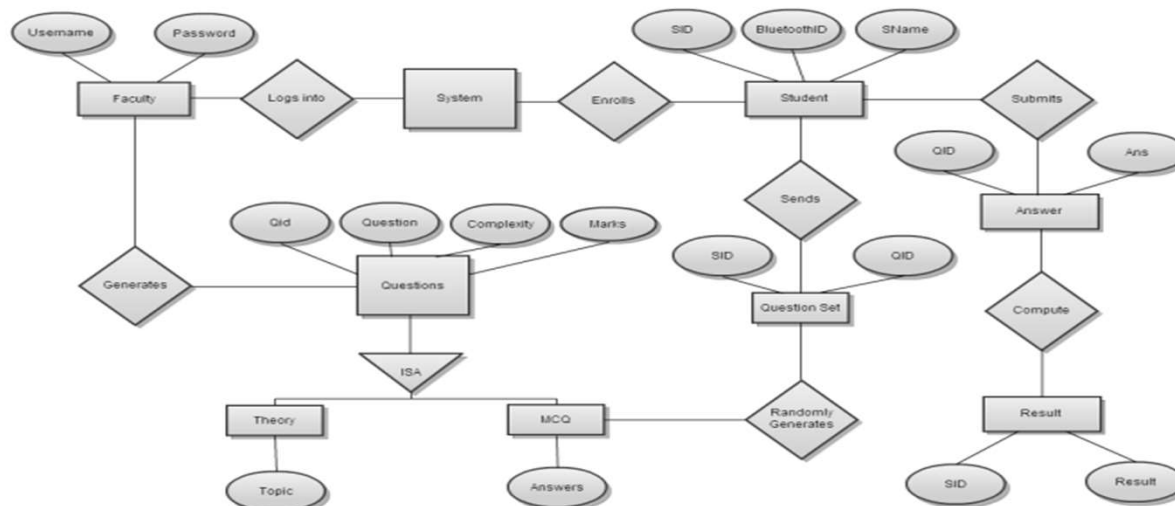
The application will be easily portable on any windows based system.

5.4 System Flowchart **Draw System Flowchart**

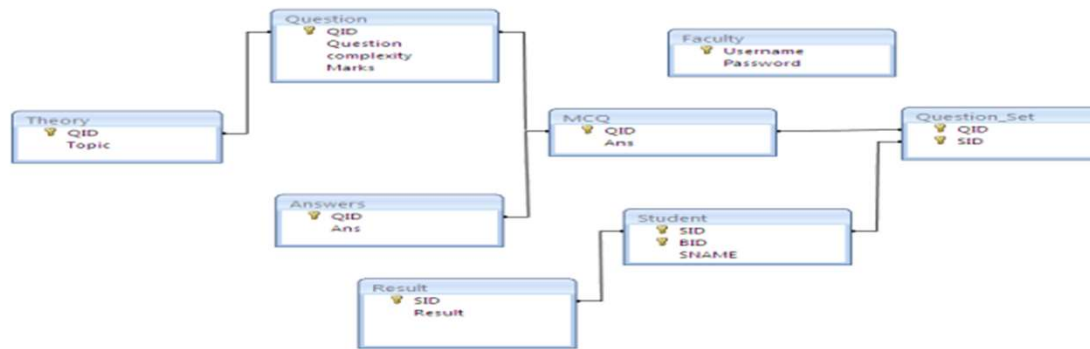
5.5 Data Flow Diagram **Draw level 0,1,2,3 DFD for system**

6. Design

6.1 E-R Diagram

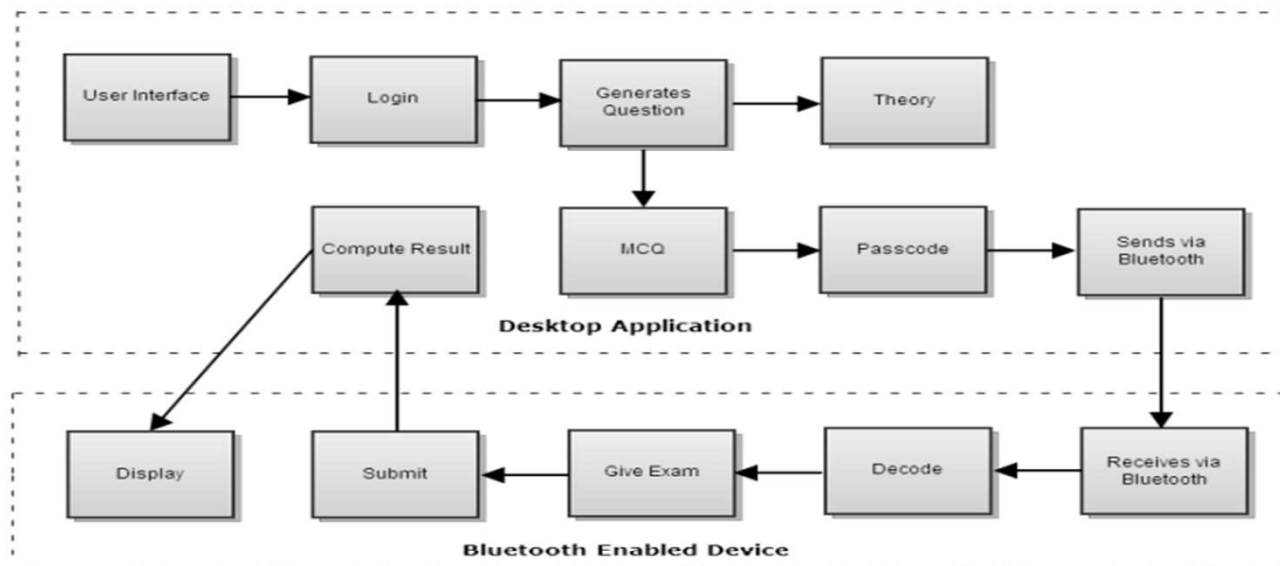


6.2 Database Schema



6.3 User Interface Design

6.3.1 System Block Diagram



6.3.2 Input Output Form

7 Other Requirements

Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse

objectives for the project, and so on. Add any new sections that are pertinent to the project

8 Glossary

Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS