K. J. Somaiya College of Engineering, Mumbai-77

(A constituent college of Somaiya Vidyavihar University)

Batch: A1	Roll No.: 16010120015 16010120013 16010120020
Experiment / assignment / tutorial No. 1	

TITLE: Requirement **S**pecification Document

AIM: To learn and understand the way of analysing the gathered information in the previous phase for the development process and prepare requirement specification document. A concept of software engineering.

Expected Course outcome of Experiment:

Process of gathering requirements and converting them into specifications.
Document created will be used by both, the customer and the developer, to understand WHAT is going to be developed.

Books/ Journals/ Websites referred:

1. Roger Pressman, Software Engineering: A practitioners Approach, McGraw Hill, 2010 ,6th edition
2. Ian Somerville, Software Engineering , Addison Wesley,2011,9th edition
3 http://en.wikipedia.org/wiki/Software_requirements_specification

Pre Lab/ Prior Concepts:

Requirements analysis in systems engineering and software engineering, encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product, taking account of the possibly conflicting requirements of the various stakeholders, such as beneficiaries or users. It is an early stage in the more general activity of requirements engineering which encompasses all activities concerned with eliciting, analyzing, documenting, validating and managing software or system requirements.

Requirements analysis is critical to the success of a systems or software project. The requirements should be documented, actionable, measurable, testable, traceable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design.

Conceptually, requirements analysis includes three types of activities:

- **Eliciting requirements**: the task of identifying the various types of requirements from various sources including project documentation, (e.g. the project charter or definition), business process documentation, and stakeholder interviews. This is sometimes also called requirements gathering.
- **Analysing requirements**: determining whether the stated requirements are clear, complete, consistent and unambiguous, and resolving any apparent conflicts.
- Recording requirements: Requirements may be documented in various forms, usually including a summary list and may include natural-language documents, use cases or process specifications.

New systems change the environment and relationships between people, so it is important to identify all the stakeholders, take into account all their needs and ensure they understand the implications of the new systems. Analysts can employ several techniques to elicit the requirements from the customer. These may include the development of scenarios, the identification of use cases, the use of workplace observation or ethnography, holding interviews, or focus groups (more aptly named in this context as requirements workshops, or requirements review sessions) and creating requirements lists. Prototyping may be used to develop an example system that can be demonstrated to stakeholders. Where necessary, the analyst will employ a combination of these methods to establish the exact requirements of the stakeholders, so that a system that meets the business needs is produced

Different types of Requirements

- Functional requirements
- Usability requirements
- Reliability requirements
- Performance requirements
- Security requirements

A typical SRS document template is shared subsequently. This document acts as a reference and will be used by both, the customer (for whom the software system is to be developed), and the organisation which develops the solution. Typically, prepared by the development organisation at the early stage of development by the professionals after interacting with the customer.

K. J. Somaiya College of Engineering, Mumbai-77

(A constituent college of Somaiya Vidyavihar University)

Software Requirements Specification for:

Spotify

Version 1.0

Prepared by 1. Rohan Dsouza

2. Khushi Kenia

3. Yash Gavade

Organisation: K.J. Somaiya College of Engineering

Date created: 26/08/2022

Table of Contents

Introduction

Spotify offers digital copyright restricted recorded music and podcasts, including more than 82 million songs, from record labels and media companies. As a freemium service, basic features are free with advertisements and limited control, while additional features, such as offline listening and commercial-free listening, are offered via paid subscriptions. Users can search for music based on artist, album, or genre, and can create, edit, and share playlists.

Purpose

Spotify is a digital music, podcast, and video streaming service that gives you access to millions of songs and other content from creators all over the world. Basic functions such as playing songs and podcasts are totally free, but you can also choose to upgrade to Spotify Premium which provides extra features and customizations.

Product Scope

Spotify is available across 182 nations, hosts 82 million+ songs, and is available to users in both free and freemium mode, with revenues generated from advertisements, and paid subscriptions.
The key feature of this platform is that it's free for users only, with some ads playing between the music. It has been found that this freemium model approach has worked amazingly for Spotify because users can take advantage of free music without paying for a subscription. Spotify makes the most of this feature to promote ads in between to keep coercing the users consciously.

Reference

https://open.spotify.com/

https://support.spotify.com/in-en

Overall Description

Product Perspective

Spotify, a self-contained product, requires system softwares depending on the individual user device. No other applications are dependent on it although third party payment gateways are used in association with it for subscription plans. Spotify originated in the music industry to

subside the illegal downloads and copyright issues regarding the ownership of songs in a billion dollar industry.

Product Functions

Spotify is a digital music, podcast, and video service that gives access to millions of songs and other content from creators all over the world.

Basic functions such as playing music are totally free, but you can also choose to upgrade to Spotify Premium.

Product functionality for a basic plan includes:

- high quality music streaming.
- creating and sharing playlists.
- customised automatic creation of playlists.

Product functionality for a premium plan includes:

- group sessions with friends up to 7 accounts to play music together.
- offline downloads.
- better quality streaming music.
- unlimited skips.
- no ads.
- all functionality included in the basic plan.

Operating Environment

Spotify Technology has an N-form organisational structure. This type of corporate structure is based on the key principle of sharing and integrating knowledge extensively throughout the music streaming enterprise and its operations.

Supported versions	
iOS	iOS 13 or above
Android	Android OS 5.0 or above
Mac	OS X 10.13 or above
Windows	Windows 7 or above

The Software is available in some cars, TVs, gaming consoles, smart watches, and more

Design and Implementation Constraints

Copyright issues are one of the unique and more difficult to manage limitations of spotify. To keep a note of what is legal and what is not and what kind of permissions and documents are required to sell a certain type of music piece. Complications might arise when the music will have to start listing albums and artists which require specific documents, prescriptions and/or legalities. The app will have to successfully allow the user to upload and share the required information/ documents, as well as properly verify them so such issues won't arise while uploading, sharing or downloading music. Hardware and software constraints will arise when devices that don't support all functions of the application are used.

User Documentation

User documentation for users is given as a step-by-step in app tutorial for the users to get accustomed to the user interface. Other sources like on-line help is provided on the official spotify website. The website is available as different domains in different countries for tutorials in other languages. Furthermore, the settings tab in the application provides a detailed explanation of what each feature provides the user.

Assumptions and Dependencies

The information about the songs provided on the software depend on the authenticity of primary data sources.The primary data sources need to be verified for them to be reliable.The songs need to be provided by a artist with identity verification and having a licence for it.

Assumptions:

- Users are expected to have a stable internet connection.
- The access to the database is fast.
- Third-party services i.e. the payment gateway
- All the information like songs artists,name,etc provided by the singer is correct.

Dependencies

- Hardware features of the server. Any problem in the server will result in the malfunctioning of the website.
- Ability of the website host to handle particular traffic on the website
- Device supports the app and all its functions.

External Interface Requirements

User Interfaces

The user interface has to be very user-friendly and intuitive, so that customers of all age groups can easily access their favourite songs from the website.

This can be made possible by following a lot of default standards that have been set by various other online platforms.

Some of the User Interface functionalities could be:

- Having an easy-to-read navigation bar/system on the homepage of the store, which will allow users to categorise and filter products based on their properties such as utility, quantity, potency, and production company.

- The website should also clearly specify the availability of product (whether a user can purchase it at the time of searching), and in the event of unavailability, also be able to accurately estimate when the product will be available again.

- When the user makes a particular search, all of the songs that match the search should be displayed on a list- based interface, and the main specifications should be shown on the listing, On clicking a particular item's listing, the user should be redirected to a webpage dedicated to that particular product, which shows the specifications of that product in detail, including photos and reviews made by previous customers.

Hardware Interfaces

No hardware is involved in the making of the music streaming platform Spotify for the time being. All of the web pages and database servers are hosted online. These external servers are services availed through other companies and are not made by developers on board the team.

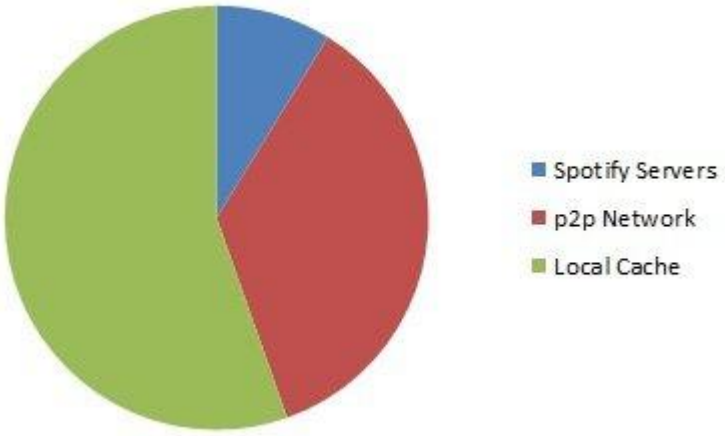
Software Interfaces

The UI for Spotify is written in displaying languages like JS, HTML, CSS. These are implemented using the Chromium Embedded Framework (CEF), but have nothing to do with their performance. What makes Spotify so incredibly fast are two things: asynchronous data loading and intelligent caching.

Most of the backend in Spotify is written in Python. When a track is played, spotify uses a very fast buffering and caching technique - this means that most of the music data comes from your computer (55%) rather than the spotify servers (9%).

Spotify Free lets you listen to music at normal (96 kilobits per second) or high quality (160 Kbps), and you can't adjust the bitrate beyond that. Spotify Premium adds Very High quality streaming at 320 Kbps, which can make for better, more detailed audio output if you use high-end headphones or speakers.

Spotify caches the music you listened to last and only has to reload less than 50% of the data when you listen to it again. Secondly, Spotify uses a clever music prediction algorithm, to find potential next songs. They will cache parts of potential music, so playback won’t have trouble buffering, even at low connection speeds.



Communications Interfaces

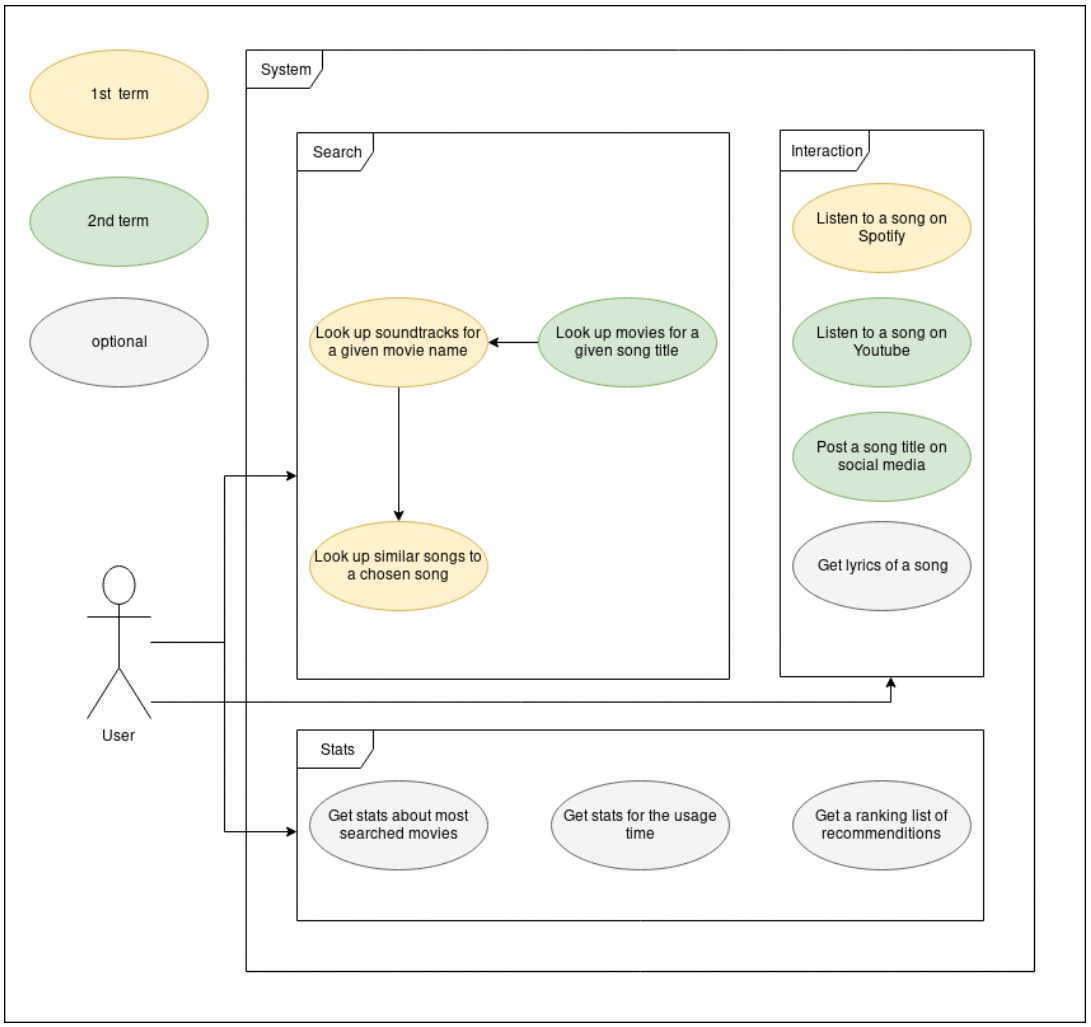
The main communication interface used is https protocol and is used for communicating over the internet since most of the functions provided in Spotify have a request response way of working. The users in turn must be able to stream songs, podcasts and videos without any delay. Sharing of songs, playlists with other users on the same platform is done as follows:
When users click the share icon, the link will direct them to the related websites i.e. Instagram, Facebook, Telegram, Email, etc which triggers the information integration process. They can also share songs via Spotify Uniform Resource Indicator (URI). This link is convenient since it directly takes users to the Spotify application, without having to go through the web page first.

System Features

Feature 1: Selecting and listening to streamed music

Introduction/Purpose of the feature:

The user shall be able to select and listen to music in on-line and mode. Spotify accesses the music via lists of playlists. During the procedure of streaming, the audio file is transmitted and delivered in small packets, which compose metafiles, and then decoded by the codec. When the buffer is filled by the decoded results, the files are turned into music and the computer straightly plays the music.



Feature 2: Making and Sharing Playlists

Playlist is the building block of Spotify. It connects different modules of Spotify, for example, from one playlist to another or from one singer to a music genre. The user can find out that playlist is everywhere in Spotify's desktop: in the home page, there displays different types of playlists in the form of square shaped photos; in the main navigation bar on the left side, there exits a list of playlists and the function of "add new playlist"; when the user search for a song, an artist or a keyword, what appears is a screen full of playlists. Playlist is the simulation of an album, in which different songs are connected by the same singer. Spotify's playlists connect songs together based on different elements, such as mood, weather, genre.

Feature 3: Group Sessions

Participants can control what is playing, and add songs to queues remotely.

- Open Spotify and play something.
- Tap the icon at the bottom of the screen.
- Tap *Start* a remote group session.
- Tap invite friends.
- If users are on the same WiFi connection, they can join the session without invitation.

Feature 4: Payment for Subscriptions

Spotify allows users to pay for a premium subscription which removes advertisements completely from the application and also lets the users download music to play it offline. Various subscription plans are available to the user including but not limited to Student plans, Family plans, Duo plans, Daily, Weekly or a year long subscription. Subscriptions are availed through online payment methods using third party payment gateways.

This project is a platform which acts as a common medium which can connect Server and the users in all areas. We have a website with good user interface for making it easy for users.

Other Nonfunctional Requirements

Performance Requirements

Some of the basic performance requirements that will be followed are:
The app/website should be running 24/7.
Payments should be stored and verified for premium plans.
Spotify Premium adds Very High quality streaming at 320 Kbps, which can make for better streams.
Database/ servers should be as active and complete the request of the customer as soon as possible.
Payments should be authenticated.
Account security should be confidential and maintain the users privacy.

Safety Requirements

Only authorised administrators should be able to query and change the contents of the database. The website users should be authenticated before they can use the feature exclusively for them. The usernames and passwords of users should be encrypted and protected from unauthorised access. Data security should be maintained. If the payments get cancelled, there must be provision for credit of the amount in question to the customer.

- **Use a secure password**
    - To keep your account as secure as possible,
- **Device security**
    - Keep your device's firmware, operating system, and/or anti-virus software up-to-date for general online security.
- **Remove access to 3rd party apps**

- If someone gains unauthorised access to your account, we recommend removing any 3rd party apps with access to your Spotify as they may have become compromised.

Security Requirements

To keep your account as secure as possible, spotify recommends you: Use a long password with letters, capitals, numbers, and special characters. Use a different password for each online service you use. Change your password frequently.

- Billing and subscriptions
- Spotify uses third party services for all payment method gateways.
- Automatic backup service can be used so that even if data is lost.
- System will use secured database
- Normal users can just read information but they cannot edit or modify anything except
- their personal and some other information.
- System will have different types of users and every user has access constraints.

Protection of Data:

Different from downloaded music, which is stored permanently in the device's hard drive and whose access does not require the connection to the internet once stored, streaming music works through wi-fi or mobile data, and the users do not "own" the music. As long as there is a steady stream of packets delivered to the computer, the user will hear the music without any interruptions.

Software Quality Attributes

What new features can be added to Spotify?

Spotify is rolling out a new feature called "Enhance," which will automatically spiff up your playlists with recommended songs that (theoretically) fit in with your music's existing style and theme. The feature works through a new Enhance button that now appears at the top of playlists.

More devices that can support the application or atleast basic versions of the same.

Increase the number of users that can join a group session remotely.

Adding contributors to playlists that can add and remove songs in a shared playlist.

Business Rules

1. Administrator can manage database and give access to other users and can view their information.
2. Users can register, login, see their own personal details, view and stream songs and purchase subscriptions.
3. Visitors can visit the site without registration and stream a certain amount of songs.
4. Sales managers can view payment details of users.

Other Requirements

Legal requirements:
Documentation for Terms of conditions of use.
Data protection appendix.
Branding guidelines.
Licences and permissions for copyright terms.

Appendix A: Glossary

User: A person that is a user of the system but has created an account.
Producers: People that upload music on spotify for users to listen.
Inventory: A database to store information about all the music and its availability in various regions.
Playlist: It is a list which contains a collection of music which can be added/removed whenever required and it is owned by a specific user.
Friends: Users who have a spotify account can follow their friends and colleagues that use spotify.

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Post Laboratory Activity:

1. You are required to prepare an SRS document for any project. (It could be the mini project you have completed in semester IV
2. Prepare a questionnaire for the allotted project considering your lab instructor is the client for requirement gathering.
3. Consider the following scenario: An institute is interested in developing a Library Information System (LIS) for the benefit of students and employees of the institute. LIS will enable the members to borrow a book (or return it) with ease while sitting at his desk/chamber. The system also enables a member to extend the date of his borrowing if no other booking for that particular book has been made. For the library staff, this system aids them to easily handle day-to-day book transactions. The librarian, who has administrative privileges and complete control over the system, can enter a new record into the system when a new book has been purchased, or remove a record in case any book is taken off the shelf. Any non-member is free to use this system to browse/search books online. However, issuing or returning books is restricted to valid users (members) of LIS only.

   The final deliverable would be a web application (using the recent HTML 5), which should run only within the institute LAN. Although this reduces security risk of the software to a large extent, care should be taken no confidential information (e.g. passwords) is stored in plain text.

   Prepare an SRS document for the same in the format discussed in the write-up.

Identification of functional requirements

The above issue explanation gives a short portrayal of the proposed framework. From the abovementioned, even without doing any profound examination, we could undoubtedly distinguish a portion of the fundamental usefulness of the framework:

New user registration: Any individual from the foundation who wishes to profit from the offices of the library needs to register himself with the LIS. On effective enrolment, a user ID and password would be given to the party.

Search book: Any individual from LIS can avail this feature to check whether a specific book is available in the establishment's library. A book could be searched by its:
- Title
- Authors name
- Publisher's name

User login: An enlisted client of LIS can login to the library by giving his user ID and password as set by him while enrolling. After effective login, "Home" page for the client is displayed from where he can get to the various functionalities of LIS: search book, issue book, return book, reissue book. Any user ID not enlisted with LIS can't get to the "Home" page - - a login disappointment message would be displayed to him, and the login exchange would show up once more. This equivalent thing happens when any enlisted user types in his password wrong. The client can respond to the security question accurately, another secret phrase would be shipped off to his/her email address. On the off chance that the client neglects to address the security question accurately, his/her LIS record would be obstructed. He really wants to contact the executive to make it dynamic once more.

Issue book: Any individual from LIS can give a book against his record gave that: a) The book is accessible in the library for example could be viewed by looking for it in LIS b) No other user has currently taken the book c) Current client has not been given the maximum number of books In the event that above conditions are met, the book is given to the user.

Return book: A book is given for a limited time frame, which we assume to be a time of 20 days. That is, a book once ought to be returned inside the following 20 days by the user from LIS. After a successful return of a book, the client account is refreshed to mirror something very similar.

Reissue book: Any user who has given a book could observe that his necessity isn't over by 20 days. All things considered, he could decide to reissue the book, and get the consent to save it for an additional 20 days. Notwithstanding, a user can reissue any book all things considered two times, after which he needs to bring it back. When a book has been effectively reissued, the user account is refreshed to mirror the data. Some other features and their priority:

- New User Registration: High
- User Login: High
- Search Book: High
- Issue Book: High
- Return Book: Medium
- Reissue Book: Medium

Identification of non-functional requirements

Having talked about functional requirements, let's try to identify a few non-functional requirements.

- Performance Requirements:
    - This system should stay open 24x7
    - At least 50 users should be able to access the system at any given time

● Security Requirements:
  ○ This system should be accessible only within the institute LAN
  ○ The database of LIS should not store any password in plain text -- a hashed value has to be stored

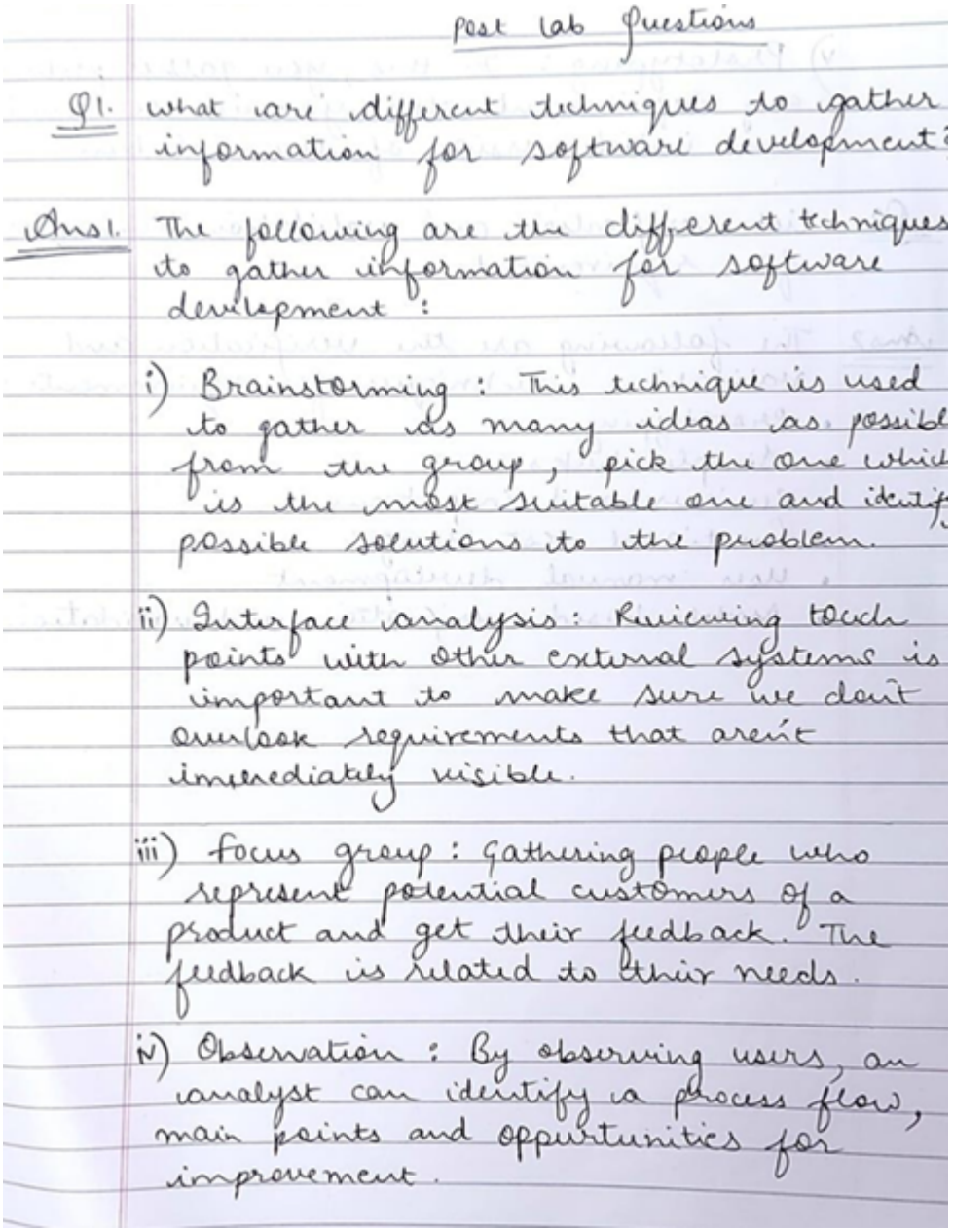
● Software Quality Attributes

● Database Requirements

● Design Constraints:
  ○ The LIS has to be developed as a web application, which should work with Firefox 5, Internet, Explorer 8, Google Chrome 12, Opera 10
  ○ The system should be developed using HTML 5.

Post Lab Descriptive Questions answers must be handwritten and to be submitted BEFORE the next term.

1. What are different techniques to gather information for software development?
2. List verification and validation techniques for requirements.



Post lab Questions

Q1. What are different techniques to gather information for software development?

Ans 1. The following are the different techniques to gather information for software development:

i) Brainstorming: This technique is used to gather as many ideas as possible from the group, pick the one which is the most suitable one and identify possible solutions to the problem.

ii) Interface analysis: Reviewing touch points with other external systems is important to make sure we don't overlook requirements that aren't immediately visible.

iii) Focus group: Gathering people who represent potential customers of a product and get their feedback. The feedback is related to their needs.

iv) Observation: By observing users, an analyst can identify a process flow, main points and opportunities for improvement.

v) Prototyping : In this, you gather preliminary requirements that you use to build the initial version of the solution.

Q2. List verification and validation techniques for requirements.

Ans 2. The following are the verification and validation techniques for requirements :

- Prototyping
- Simple checks
- Reviews and inspections
- functional test design
- User manual development
- Model- based verification and validation