Software Requirement Specification

Version 1.0

Date:- 31 October 2018

### Quikr Clone

Simran Sharma

(1611981375)

Submitted in partial fulfillment

of the requirements of

CSL3306 Software Engineering

To

Ms. Tazeen

TABLE OF CONTENTS

1. Introduction
   1. Purpose
   2. Scope
   3. Definition, Acronyms or abbreviations
   4. References
   5. Overview
2. General Description
   1. Product Perspective
   2. Product Functions
   3. User Characteristics
   4. General Constraints
   5. Assumptions
3. Functional Requirements
4. Non Functional Requirements
5. System Architecture
6. System Models
7. Appendices

INTRODUCTION

Quikr provides an online [classified advertising](https://en.wikipedia.org/wiki/Classified_advertising) platform for users to buy or sell goods and services from each other. Other services offered include a [missed call](https://en.wikipedia.org/wiki/Missed_call) service and [instant messaging](https://en.wikipedia.org/wiki/Instant_messaging),

# PURPOSE

This project aims to provide a similar platform, \_\_\_\_\_\_\_\_, to help the students familiarize themselves with the various features that Java and HTML provide for web and app designing.

# SCOPE

This project particularly aims towards improving the ways in which students are evaluated and tested. Furthermore, both teachers and parents of a particular student can easily look upon the progress of the student.

Moreover, this project can subvert and help students to improve and shuffle them on to the next level.

# DEFINITION, ACRONYMS, or ABBREVIATIONS

|  |  |
| --- | --- |
| **TERM** | **DEFINITION** |
| Author | Person submitting an article to be reviewed. In case of multiple authors, this term refers to the *principal author*, with whom all communication is made. |
| Database | Collection of all the information monitored by this system. |
| Software Requirements Specification(SRS) | A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document. |
| Reader | Anyone visiting the site to read articles. |
| Field | A cell within a form. |
| HTML (Hyper Text Mark-Up Language) | Markup Language used to create web-pages. |
| MySQL | Database management system used for the project. |
| Java | Object oriented programming language. |
| JDBC | Java Database Connectivity. |
| JSP | Java Servlet Programming. |
| User | Reviewer or Author. |

# REFERENCES

# IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

# OVERVIEW

# In the next section, the General Description section, of this document gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter.

# The third and fourth sections, functional requirements and non-functional requirements, which are requirements specific section, of this document is written primarily for the developers and describes in technical terms the details of the functionality of the product.

# Fifth and sixth chapters focus on the system architecture and system models which define the structure, behaviour, and modular views of a system.

# Both sections of the document describe the same software product in its entirety, but are intended for different audiences and thus use different languages.

GENERAL DESCRIPTION

# PROJECT PERSPECTIVE

# In the modern era of digitisation, the newspapers and advertising sector has also stepped up. Advertising, buying and selling of goods and services have now come into the digital world.

Digitisation has its pros and cons for the advertising sector and in this project we intend to expand the list of pros while finding solutions and ways to work around the cons. This main objective of this SRS is to present our strategies, vision and goals when we talk about electronic learning independently of is topology.   
PROJECT FUNCTIONS

This project aims to create a user friendly platform to:

Sell goods in a fairly good shape

Buy good at a nominal second-hand rate

# USER CHARACTERISTICS

This product is intended for use by the people who’ve bought something they were once passionate about to sell it to someone who is now passionate about it. It offers goods at an affordable price and good condition.

# GENERAL CONSTRAINTS

# This product will work on client-server architecture. It will be require an internet server and which will be able to run Java application. The product should be support some commonly used browsers such as Internet Explorer, Mozilla Firefox. External interfaces include key board and mouse, enabling navigations across the screens.

# ASSUMPTIONS

* All the users (ie students and teachers) have a scintillating internet connection.
* The Desktops/ Laptops and other hardware requirements are already present with the user.
* JRE/ JVM are already installed in the computer systems.
* All systems must support atleast dll 365 netframework beta.

FUNCTIONAL REQUIREMENTS

This product is designed with the aim to facilitate communication between two types of end users -the buyer and the seller – of an item from one corner of the world to another.

This product aims to allow a user to view other items available for sale. It also aims to support and answer queries put forth by various buyers for an item while allowing a private chat between the buyers and the seller.

NON FUNCTIONAL REQUIREMENTS

HARDWARE REQUIREMENTS

The application demands that all the PCs must be present in the internet. The PC should be sufficiently fast with adequate memory at least 64 MB RAM and 2 GB hard –disk space is required to run this application. Screen resolution of at least 800\*600 required to properly view the screen. It should also be able to support the printers

# SOFTWARE REQUIREMENTS

Any Window Operating System(not DOS).

Java must be installed. For the database handling MYSQL and SQLite manager must be installed.

The final application must be packaged in a set up program, so that the product can be easily installed on the clients-machine.

OTHER REQUIREMENTS

### Security

### Application will allow only valid users to access the system. Access to any will application resource depend upon user’s designation. There are two types

of users namely Administrator and Buyer/Seller . Security is based upon the individual user ID and Password.

### Reliability

The application should be highly reliable and it should generate all the updated information in correct order.

### Availability

### System will be available around the clock except for the time required for the back up of data.

### Maintainability

The installation and operation manual of student management system will be provided to user.

### Portability

The application should be portable on any windows based system.

SYSTEM ARCHITECTURE

# FRAMEWORKS USED

### Java

Java is a programming language created by James Gosling from Sun Microsystems (Sun) in 1991. The target of Java is to write a program once and then run this program on multiple operating systems. The first publicly available version of Java (Java 1.0) was released in 1995. Sun Microsystems was acquired by the Oracle Corporation in 2010. Oracle has now the steermanship for Java. In 2006 Sun started to make Java available under the GNU General Public License (GPL). Oracle continues this project called *OpenJDK*.

Over time new enhanced versions of Java have been released. The current version of Java is Java 1.8 which is also known as *Java 8*.

Java is defined by a specification and consists of a programming language, a compiler, core libraries and a runtime (Java virtual machine) The Java runtime allows software developers to write program code in other languages than the Java programming language which still runs on the Java virtual machine. The *Java platform* is usually associated with the *Java virtual machine* and the *Java core libraries*.

The Java language was designed with the following properties:

Platform independent: Java programs use the Java virtual machine as abstraction and do not access the operating system directly. This makes Java programs highly portable. A Java program (which is standard-compliant and follows certain rules) can run unmodified on all supported platforms, e.g., Windows or Linux.

Object-orientated programming language: Except the primitive data types, all elements in Java are objects.

Strongly-typed programming language: Java is strongly-typed, e.g., the types of the used variables must be pre-defined and conversion to other objects is relatively strict, e.g., must be done in most cases by the programmer.

Interpreted and compiled language: Java source code is transferred into the bytecode format which does not depend on the target platform. These bytecode instructions will be interpreted by the Java Virtual machine (JVM). The JVM contains a so called Hotspot-Compiler which translates performance critical bytecode instructions into native code instructions.

Automatic memory management: Java manages the memory allocation and de-allocation for creating new objects. The program does not have direct access to the memory. The so-called garbage collector automatically deletes objects to which no active pointer exists.

The Java syntax is similar to C++. Java is case-sensitive, e.g., variables called myValue and myvalue are treated as different variables.

### JDBC (Java Data Base Connectivity)

The JDBC API is a Java API that can access any kind of tabular data, especially data stored in a Relational Database.

JDBC helps you to write Java applications that manage these three programming activities:

Connect to a data source, like a database

Send queries and update statements to the database

Retrieve and process the results received from the database in answer to your query

JDBC includes four components:

**The JDBC API** —  The JDBC API provides programmatic access to relational data from the Java programming language. Using the JDBC API, applications can execute SQL statements, retrieve results, and propagate changes back to an underlying data source. The JDBC API can also interact with multiple data sources in a distributed, heterogeneous environment.

The JDBC API is part of the Java platform, which includes the Java Standard Edition (Java SE ) and the Java Enterprise Edition (Java EE). The JDBC 4.0 API is divided into two packages: java.sql and javax.sql. Both packages are included in the Java SE and Java EE platforms.

**JDBC Driver Manager** —  The JDBC DriverManager class defines objects which can connect Java applications to a JDBC driver. DriverManager has traditionally been the backbone of the JDBC architecture. It is quite small and simple.

The Standard Extension packages javax.naming and javax.sql let you use a DataSource object registered with a Java Naming and Directory Interfaces (JNDI) naming service to establish a connection with a data source. You can use either connecting mechanism, but using a DataSource object is recommended whenever possible.

**JDBC Test Suite** —  The JDBC driver test suite helps you to determine that JDBC drivers will run your program. These tests are not comprehensive or exhaustive, but they do exercise many of the important features in the JDBC API.

**JDBC-ODBC Bridge** —  The Java Software bridge provides JDBC access via ODBC drivers. Note that you need to load ODBC binary code onto each client machine that uses this driver. As a result, the ODBC driver is most appropriate on a corporate network where client installations are not a major problem, or for application server code written in Java in a three-tier architecture.

This Trail uses the first two of these these four JDBC components to connect to a database and then build a java program that uses SQL commands to communicate with a test Relational Database. The last two components are used in specialized environments to test web applications, or to communicate with ODBC-aware DBMSs.

### ASP

Java Server Pages (JSP) is a server-side programming technology that enables the creation of dynamic, platform-independent method for building Web-based applications. JSP have access to the entire family of Java APIs, including the JDBC API to access enterprise databases.

JavaServer Pages often serve the same purpose as programs implemented using the Common Gateway Interface (CGI). But JSP offers several advantages in comparison with the CGI.

Performance is significantly better because JSP allows embedding Dynamic Elements in HTML Pages itself instead of having separate CGI files.

JSP are always compiled before they are processed by the server unlike CGI/Perl which requires the server to load an interpreter and the target script each time the page is requested.

JavaServer Pages are built on top of the Java Servlets API, so like Servlets, JSP also has access to all the powerful Enterprise Java APIs, including JDBC, JNDI, EJB, JAXP, etc.

JSP pages can be used in combination with servlets that handle the business logic, the model supported by Java servlet template engines.

Finally, JSP is an integral part of Java EE, a complete platform for enterprise class applications. This means that JSP can play a part in the simplest applications to the most complex and demanding.

APPENDICES

This project can be expanded from just old goods to services.

It may go on to include saloon services, household services like electricians, plumbers, carpenters. It could also provide a platform for upcoming artists to display and sell their work of art from any corner of the world to an admirer in a different corner of the world.