**Book Management system -App**

**Introduction**

The aim of this Book Management System is the management of a bookstore library. It will facilitate the process of finding and checking out books from libraries. Libraries generally hold books numbering in the tens of thousands and a single person will find it daunting to simply find out if a certain book is available. A database of the books in a library will be maintained. A librarian can update this database using the Admin module. This database will carry important information about the book such as the title, author’s name, date of publication, publisher, and ISBN (International Standard Book Number). Each book will be assigned a unique serial number and an ID and stored in DB. Any registered user can rent the book with details for a certain period, User can see the book and if logged in they can pick the book. When renting the book he must provide the return date ID card number and ID photo

**Abstract**

When a user checks out a book, the librarian will access the check–out form, and the book details have to be entered into the system, it will automatically store the date and time of check–out, ID details of the user checking out the book and the serial number of the book, the system will also automatically reduce the number of available copies of the book on the book database. When a user returns the book, the system will check if the book return was delayed and then it will update the book database to indicate the return of the book to the library.

**Existing System**

The existing system is very slow and inefficient. Managing multiple data is also not an easy task in the current situation. There is a lot of manual work involved in the current system and mistakes in one detail can lead to a wrong entry in the system. No proper collection of requirements leads to a huge problem for this system. This system is to enhance manual work and also more energy is wasted on arranging the report generation format.

It is difficult to find out how many copies of the books remain and calculating any delays in book return is a time-consuming process.

**Disadvantages of Existing System**

The current system is manual so all the records are maintained manually. So arranging the data securely and generating result details cannot be determined if a new entry is added to the list

Time-Consuming

Less Efficient

More manual Work Required

Less Accurate

Not User Friendly

**Proposed System**

1. This system is user-friendly for the retrieval and storing of data. And it is fast to store the data. It is maintained efficiently. The graphical user interface is implemented in this proposed system. It is more efficient than the existing system.

2. Reports are easily generated like table lists can be easily generated in this proposed system so that maintenance is straightforward

3. Very Less PaperWork, The proposed system requires very little paperwork. All the data is entered into the computer immediately and reports can be generated with the help of computers. That work will become very easy because there is no need to keep data on more papers.

4. Computer Operator Control Computer operator control is available so the rate of errors will be less. Storing and retrieving information is simple. So work can be done at the correct time and also at a good speed.

5 The system also enables users to find out about the availability of books they require and reserve copies.

**Modules**

**Admin:** The admin module will be able to edit the database of books. Books can be added, or deleted or the information can be edited. The admin module will also be able to view statistics of book issuing, this will help the library make future decisions about which books to purchase and in what quantity. The admin module is also responsible for creating login IDs for new users.

**Librarian:** The librarian module can access the book checkout and book return forms. They will be able to update the book details using these forms.

**User:** The user module can be used to access information about book availability and reservation of book copies

**Requirements**

**Software Requirements**

Operating system: Windows 10.

Coding Language: Android ( JAVA)

IDE: Android Studio

Database: SQLite

**Hardware Minimum Requirements**

System: Pentium Processor 2.4 GHz.

Hard Disk: 80 GB.

Monitor: 15 VGA Colour.

Ram: 4 GB

**System Design**

Design is the first step in the development phase for any techniques and principles to define a device, a process, or a system in sufficient detail to permit its physical realization.

Once the software requirements have been analyzed and specified the software design involves three technical activities - design, coding, implementation, and testing that are required to build and verify the software.

The design activities are of main importance in this phase, because in this activity, decisions ultimately affecting the success of the software implementation and its ease of maintenance are made. These decisions have the final bearing on the reliability and maintainability of the system. Design is the only way to accurately translate the customer’s requirements into finished software or a system.

Design is the place where quality is fostered in development. Software design is a process through which requirements are translated into a representation of software. Software design is conducted in two steps. Preliminary design is concerned with the transformation of requirements into data.

**Input Design**

The input design is the link between the information system and the user. It comprises the developing specifications and procedures for data preparation and those steps are necessary to put transaction data into a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps, and keeping the process simple. The input is designed in such a way so that it provides security and ease of use while retaining privacy. Input Design considered the following things:

What data should be given as input?

How the data should be arranged or coded?

The dialog guides the operating personnel in providing input.

Methods for preparing input validations and steps to follow when errors occur.

**Output Design**

A quality output is one, which meets the requirements of the end user and presents the information. In any system results of processing are communicated to the users and to another system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source of information to the user. Efficient and intelligent output design improves the system’s relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well-thought-out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can be used easily and effectively. When analyzing design computer output, they should Identify the specific output that is needed to meet the requirements.

2. Select methods for presenting information.

3. Create documents, reports, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

Convey information about past activities, current status or projections of the

Future.

Signal important events, opportunities, problems, or warnings.

Trigger an action.

Confirm an action.

**Tables**

**UserInfo**

| FieldName | Type |
| --- | --- |
| Id | int |
| Name | string |
| EmailId | string |
| Phone | string |
|  |  |

**BookInfo**

| FieldName | Type |
| --- | --- |
| Id | int |
| Name | string |
| BookCode | string |
| Section | string |
| LibraryCode | string |

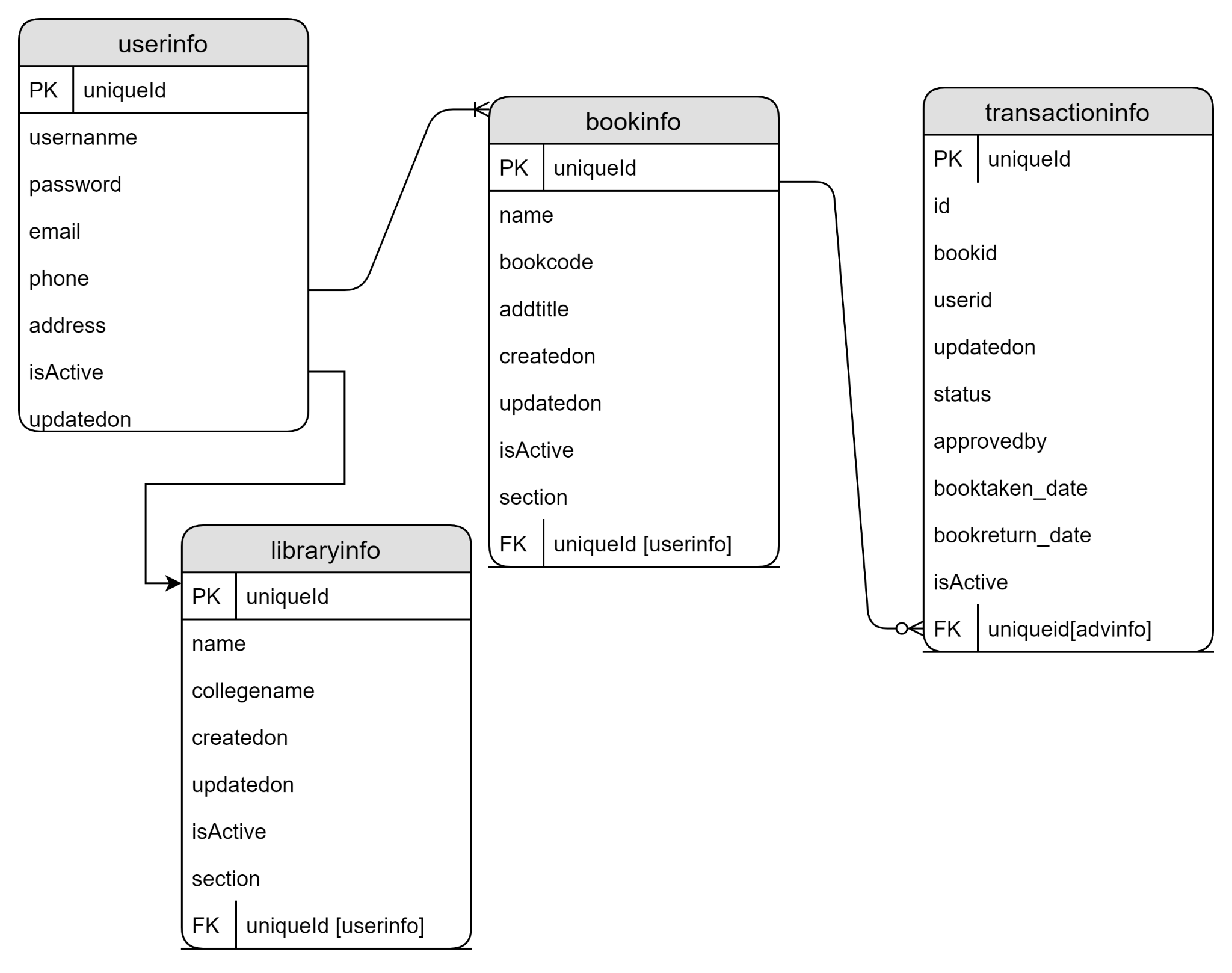
**LibraryInfo**

| FieldName | Type |
| --- | --- |
| Id | int |
| Name | string |
| Collegename | string |
| LibraryCode | string |
|  |  |

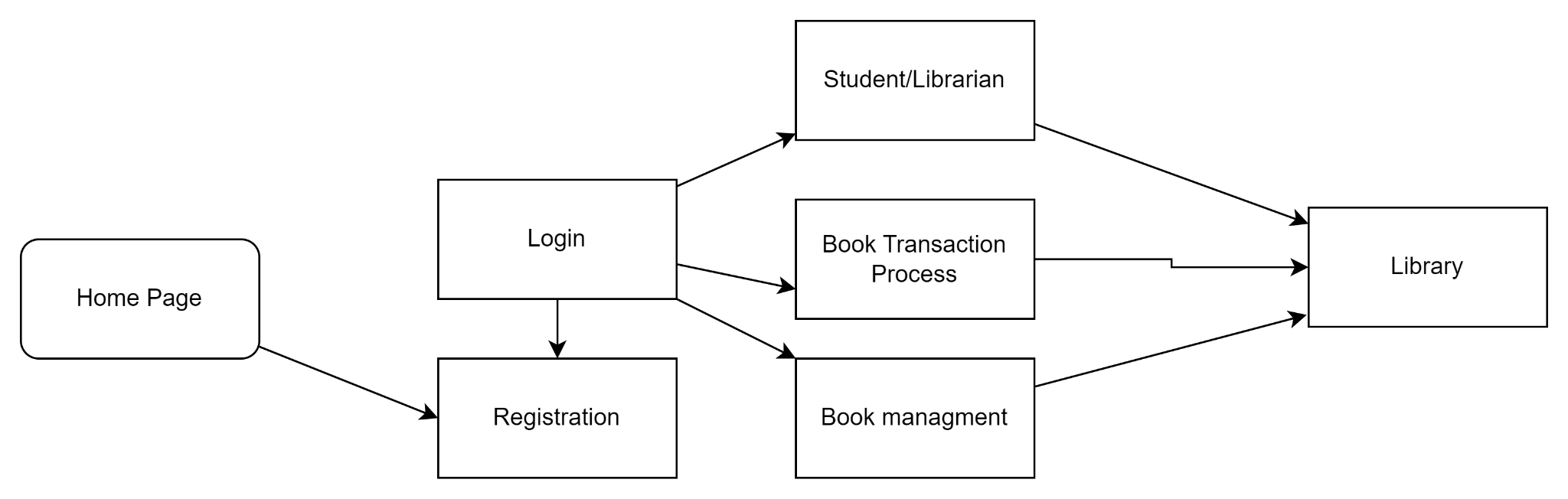
**BookTransactionInfo**

| FieldName | Type |
| --- | --- |
| Id | int |
| studentid | int |
| librarianid | int |
| booktaken\_date | datetime |
| bookreturn\_date | datetime |
| isdateExceeded | boolean |
|  |  |

**Database Normalization Diagram**



**System Architecture**



**ENTITY RELATIONSHIP DIAGRAM**































**DFD**

**Level 0**







**Level 1**













Process



**Level 2**





























**SYSTEM FLOW DIAGRAM**





Invalid



Valid





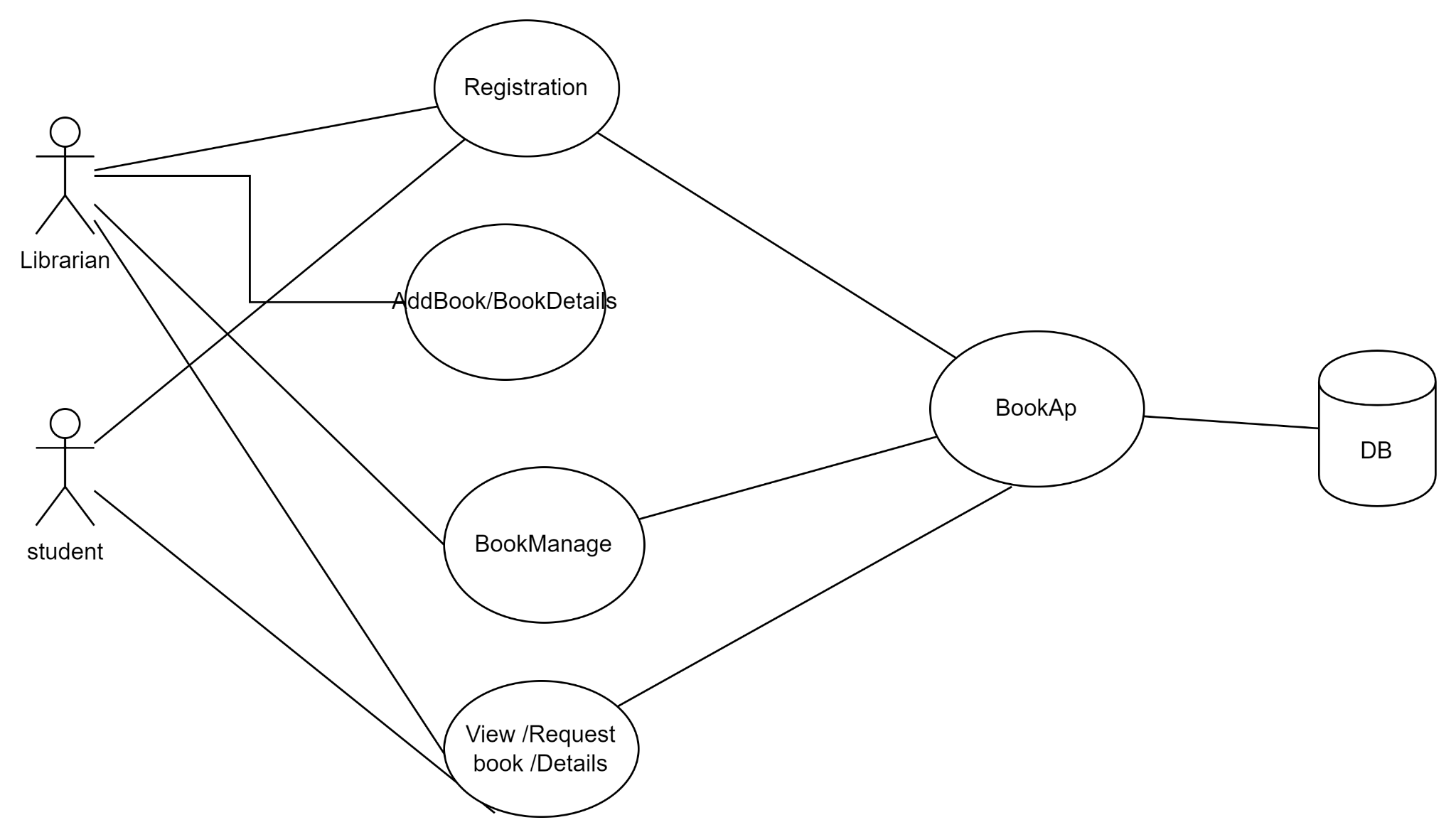








**Use Case Diagram**



**Technology**

**Background Study**

* **Background study**

**ANDROID:**

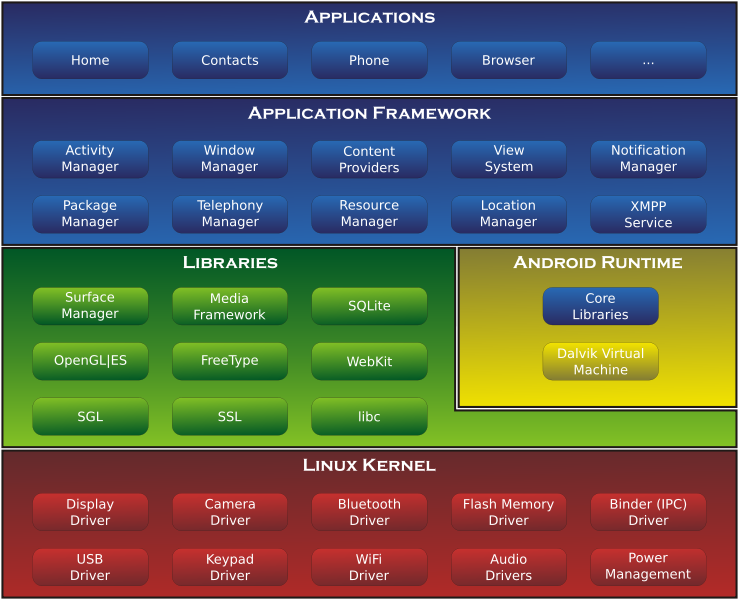
Android is a Linux-based operating system designed primarily for touch screen mobile devices such as smart phones and tablet computers, developed by Google in conjunction with the Open Hand-set Alliance. Initially developed by Android Inc, whom Google financially backed and later purchased in 2005, Android was unveiled in 2007 along with the founding of the Open Handset Alliance, a con-sortium of 86 hardware, software, and telecommunication companies devoted to advancing open standards for mobile devices.

Google releases the Android code as open-source, under the Apache License. The Android Open Source Project (AOSP), lead by Google, is tasked with the maintenance and further development of Android. Additionally, Android has a large community of developers writing applications ("apps") that extend the functionality of devices. Developers write primarily in a customized version of Java, and apps can be downloaded from online stores such as Google Play (formerly Android Mar-ket), the app store run by Google, or third-party sites.

The user interface in Android is based on the concept of direct manipulation, using single- and multi-touch inputs like swiping, tapping, pinching and reverse pinching to manipulate on-screen objects. The response to user input is designed to be immediate and provides a fluid touch interface. In-ternal hardware such as accelerometers, gyroscopes and proximity sensors are used by some applications to respond to additional user actions, for example adjusting the screen from portrait to landscape de-pending on how the device is orientated, or allowing the user to steer a vehicles in a racing game by rotating the device, simulating control of a steering wheel.

Android devices boot to the home screen, which is similar to the desktop found on PCs, and is the primary navigation and information point on the device. Android home screens are typically made up of app icons, which launch the associated app, and widgets, which display live, auto-updating con-tent such as the weather forecast, the user's email inbox, or a news ticker directly on the home screen. A home screen may be made up of several pages that the user can swipe back and forth between.

**ARCHITECTURE DIAGRAM:**

****

**Android Open Source Project**

The Android Open Source Project is led by Google, and tasked with the maintenance and development of Android. According to the project "The goal of the Android Open Source Project is to create a successful real-world product that improves the mobile experience for end users" AOSP also maintains the Android Compatibility Program, defining an "Android compatible" device "as one that can run any application written by third-party developers using the [Android SDK](http://en.wikipedia.org/wiki/Android_SDK) and [NDK](http://en.wikipedia.org/wiki/Android_NDK)", to prevent incompatible Android implementations. The compatibility program is optional and free of charge, with the Compatibility Test Suite also free and open-source.

**APPLICATIONS:**

### Applications can be acquired by end-users either through a store such as Google Play or the Amazon Appstore, or by downloading and installing the application's APK file from a third-party site. Google Play

****

Google Play is an online software store developed by Google for Android devices. An application program ("app") called "Play Store" is preinstalled on most Android devices and allows users to browse and download apps published by third-party developers, hosted on Google Play. As of June 2012, there were more than 600,000 apps available for Android, and the estimated number of applications downloaded from the Play Store exceeded 20 billion. The operating system itself is installed on 400 million total devices.

Only devices that comply with Google's compatibility requirements are allowed to preinstall and access the Play Store. The app filters the list of available applications to those that are compatible with the user's device, and developers may restrict their applications to particular carriers or countries for business reasons.

Google offers many free applications in the Play Store including Google Voice, Google Goggles, Gesture Search, Google Translate, Google Shopper, Listen and My Tracks. In August 2010, Google launched "Voice Actions for Android", which allows users to search, write messages, and initiate calls by voice.

Security:

Android applications run in a sandbox, an isolated area of the operating system that does not have access to the rest of the system's resources, unless access permissions are granted by the user when the application is installed. Before installing an application, the Play Store displays all required permissions. A game may need to enable vibration, for example, but should not need to read messages or access the phonebook. After reviewing these permissions, the user can decide whether to install the application. The sandboxing and permissions system weakens the impact of vulnerabilities and bugs in applications, but developer confusion and limited documentation has resulted in applications routinely requesting unnecessary permissions, reducing its effectiveness. The complexity of inter-application communication implies Android may have opportunities to run unauthorized code.

Privacy:

Android smart phones have the ability to report the location of Wi-Fi access points, encountered as phone users move around, to build databases containing the physical locations of hundreds of millions of such access points. These databases form electronic maps to locate smart phones, allowing them to run apps like Foursquare, Latitude, Places, and to deliver location-based ads. Third party monitoring software such as TaintDroid, an academic research-funded project, can, in some cases, detect when personal information is being sent from applications to remote servers.

Open source community:

Android has an active community of developers and enthusiasts who use the Android source code to develop and distribute their own modified versions of the operating system. These community-developed releases, the most widely used of which being Cyanogen Mod, often bring new features and updates to devices faster than through the official manufacturer/carrier channels, albeit without as extensive testing or quality assurance. Community releases often come pre-rooted and contain modifications unsuitable for non-technical users, such as the ability to over clock or over/under volt the device's processor.

Historically, the early responses of tablet and smart phone manufacturers and mobile carriers were typically unsupportive of third-party firmware development. Manufacturers expressed concern about improper functioning of devices running unofficial software and the support costs resulting from this. Moreover, modified firmwares such as Cyanogen Mod sometimes offer features for which carriers would otherwise charge a premium (e.g., tethering). As a result, technical obstacles including locked bootloaders and restricted access to root permissions were common in many devices. However, as community-developed software has grown more popular, and following a statement by the Librarian of Congress in the United States that permits the "jailbreaking" of mobile devices, manufacturers and carriers have softened their position regarding third party development, with some, including HTC, Motorola, Samsung and Sony Ericsson, providing support and encouraging development. As a result of this, over time the need to circumvent hardware restrictions to install unofficial firmware has lessened as an increasing number of devices are shipped with unlocked or unlockable bootloaders, similar to the Nexus series of phones, although usually requiring that users waive their devices' warranties to do so.

Features:

* Speed, memory, and performance optimizations.
* Additional application speed improvements, implemented through [JIT compilation](http://en.wikipedia.org/wiki/Just-in-time_compilation).
* Integration of [Chrome](http://en.wikipedia.org/wiki/Google_Chrome)'s [V8 JavaScript engine](http://en.wikipedia.org/wiki/V8_(JavaScript_engine)) into the Browser application
* Support for the [Android Cloud to Device Messaging](http://en.wikipedia.org/wiki/Android_Cloud_to_Device_Messaging) (C2DM) service, enabling [push notifications](http://en.wikipedia.org/wiki/Push_technology).
* Improved Microsoft Exchange support, including security policies, auto-discovery, GAL look-up, calendar synchronization and remote wipe
* Improved application launcher with shortcuts to Phone and Browser applications
* USB tethering and [Wi-Fi hotspot](http://en.wikipedia.org/wiki/Wi-Fi_hotspot) functionality
* Added an option to disable data access over mobile network
* Updated Market application with batch and automatic update features.
* Quick switching between multiple keyboard languages and their dictionaries
* Voice dialing and contact sharing over Bluetooth.
* Support for Bluetooth-enabled car and desk docks.
* Support for numeric and alphanumeric passwords.
* Support for file upload fields in the Browser application.
* Support for installing applications to the expandable memory.
* [Adobe Flash](http://en.wikipedia.org/wiki/Adobe_Flash) support.
* Support for extra-high-[PPI](http://en.wikipedia.org/wiki/Pixel_density) screens (320 ppi), such as 4" 720p.
* Gallery allows users to view picture stacks using a zoom gesture

**ANDROID VERSION:**

The Nexus Q is a media-streaming entertainment device in the Google Nexus product family. The device runs version 5.0 of the Android operating system, Lollypop and integrates with Google Play. The device has a spherical shape.

Features

The Nexus Q can be connected to an HDTV or home stereo system to play digital audio and video content from Google Play or YouTube. The device runs a custom version of Google's Android operating system, based on Android 5.0 Lollypop. The device can be controlled by smart phones and tablets running the Android operating system.

Content

All content displayed by the Nexus Q is streamed over the internet directly from Google Play, no content is stored locally. The device does contain a reported 16GB of internal flash-based storage, however, as is the case with Apple TV, internal storage is simply used for temporarily storing buffered digital content as it is being streamed to the device over the internet from Google Play servers.

### **Hardware Design**

The Nexus Q is a unique spherical shape, with a ring of LEDs running around the circumference of the device separating it into two halves. While music is playing, the LEDs change in correspondence with the music waveforms. If the Q is attached to a TV, it can display a visualize as well. The top half of the device can be rotated to change the audio volume being output over attached speakers or to other home theater equipment. The surface of the device also contains a capacitive-touch sensitive button for tog-gling an audio mute feature.

The back of the Q contains an Ethernet port for attaching to terrestrial home networks. There is also micro-HDMI port for connecting to HDTVs and other home theater equipment for outputting digital audio and video. A S/PDIF optical digital audio connector is included for outputting digital audio to home theater receivers and other compatible stereo equipment.

A micro-USB port was also included officially for service and support purposes, however, during its unveiling during the Google I/O 2012 keynote address, Google Director of Hardware Matt Hershenson announced that it was also to "connect future accessories and encourage general hack-ability." In addition to the digital connections, the Q also includes a row of four analog banana plug audio jacks for attaching directly to a pair of stereo speakers. The Q itself includes a "stereo-grade" amplifier for outputting audio directly to analog speakers.

### **Wireless**

The Nexus Q utilizes many different wireless technologies for connecting with content services and other devices. The device can be controlled by an Android Smartphone or tablet over Wi-Fi (dual-band), Bluetooth, or NFC. A demo at Google I/O 2012 showed Samsung Galaxy Nexus and Nexus 7 users connecting to the same wireless network the Q was connected to, and selecting music and video to be played by the Q. This functionality is similar to Apple's Airplay technology, however, unlike Air-play, where content is streamed directly from the controlling device, an Android device merely tells the Q what content to play, and then the content is streamed directly from Google Play to the Q over the internet.

**MySQL**

A database is a separate application that stores a collection of data. Each database has one or more distinct APIs for creating, accessing, managing, searching and replicating the data it holds.

Other kinds of data stores can be used, such as files on the file system or large hash tables in memory but data fetching and writing would not be so fast and easy with those types of systems.

So nowadays, we use relational database management systems (RDBMS) to store and manage huge volume of data. This is called relational database because all the data is stored into different tables and relations are established using primary keys or other keys known as foreign keys.

A **Relational Database Management System (RDBMS)** is a software that:

* Enables you to implement a database with tables, columns and indexes.
* Guarantees the Referential Integrity between rows of various tables.
* Updates the indexes automatically.
* Interprets an SQL query and combines information from various tables.

## RDBMS Terminology:

Before we proceed to explain MySQL database system, let's revise few definitions related to database.

* **Database:** A database is a collection of tables, with related data.
* **Table:** A table is a matrix with data. A table in a database looks like a simple spreadsheet.
* **Column:** One column (data element) contains data of one and the same kind, for example the column postcode.
* **Row:** A row (= tuple, entry or record) is a group of related data, for example the data of one subscription.
* **Redundancy:** Storing data twice, redundantly to make the system faster.
* **Primary Key:** A primary key is unique. A key value cannot occur twice in one table. With a key, you can find at most one row.
* **Foreign Key:** A foreign key is the linking pin between two tables.
* **Compound Key:** A compound key (composite key) is a key that consists of multiple columns, because one column is not sufficiently unique.
* **Index:** An index in a database resembles an index at the back of a book.
* **Referential Integrity:** Referential Integrity makes sure that a foreign key value always points to an existing row.

## MySQL Database:

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed, and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons:

* MySQL is released under an open-source license. So you have nothing to pay to use it.
* MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
* MySQL uses a standard form of the well-known SQL data language.
* MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
* MySQL works very quickly and works well even with large data sets.
* MySQL is very friendly to PHP, the most appreciated language for web development.
* MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).
* MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

**SQL Lite Database**

SQLite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. The code for SQLite is in the public domain and is thus free for use for any purpose, commercial or private. SQLite is the most widely deployed database in the world with more applications than we can count, including several high-profile projects.

SQLite is an embedded SQL database engine. Unlike most other SQL databases, SQLite does not have a separate server process. SQLite reads and writes directly to ordinary disk files. A complete SQL database with multiple tables, indices, triggers, and views, is contained in a single disk file. The database file format is cross-platform - you can freely copy a database between 32-bit and 64-bit systems or between big-endian and little-endian architectures. These features make SQLite a popular choice as an Application File Format. SQLite database files are a recommended storage format by the US Library of Congress. Think of SQLite not as a replacement for Oracle but as a replacement for fopen()

SQLite is a compact library. With all features enabled, the library size can be less than 750KiB, depending on the target platform and compiler optimization settings. (64-bit code is larger. And some compiler optimizations such as aggressive function inlining and loop unrolling can cause the object code to be much larger.) There is a tradeoff between memory usage and speed. SQLite generally runs faster the more memory you give it. Nevertheless, performance is usually quite good even in low-memory environments. Depending on how it is used, SQLite can be faster than direct filesystem I/O.

SQLite is very carefully tested prior to every release and has a reputation for being very reliable. Most of the SQLite source code is devoted purely to testing and verification. An automated test suite runs millions and millions of test cases involving hundreds of millions of individual SQL statements and achieves 100% branch test coverage. SQLite responds gracefully to memory allocation failures and disk I/O errors. Transactions are ACID even if interrupted by system crashes or power failures. All of this is verified by the automated tests using special test harnesses which simulate system failures. Of course, even with all this testing, there are still bugs. But unlike some similar projects (especially commercial competitors) SQLite is open and honest about all bugs and provides bugs lists and minute-by-minute chronologies of code changes.

The SQLite code base is supported by an international team of developers who work on SQLite full-time. The developers continue to expand the capabilities of SQLite and enhance its reliability and performance while maintaining backwards compatibility with the published interface spec, SQL syntax, and database file format. The source code is absolutely free to anybody who wants it, but professional support is also available.

The SQLite project was started on 2000-05-09. The future is always hard to predict, but the intent of the developers is to support SQLite through the year 2050. Design decisions are made with that objective in mind.

**SYSTEM TESTING:**

Testing is performed to identify errors. It is used for quality assurance. Testing is an integral part of the entire development and maintenance process. The goal of the testing during phase is to verify that the specification has been accurately and completely incorporated into the design, as well as to ensure the correctness of the design itself. For example the design must not have any logic faults in the design is detected before coding commences, otherwise the cost of fixing the faults will be considerably higher as reflected. Detection of design faults can be achieved by means of inspection as well as walkthrough.

Testing is one of the important steps in the software development phase. Testing checks for the errors, as a whole of the project testing involves the following test cases:

Ø Static analysis is used to investigate the structural properties of the Source code.

Ø Dynamic testing is used to investigate the behavior of the source code by executing the program on the test data.

**1. UNIT TESTING:**

Unit testing is conducted to verify the functional performance of each modular component of the software. Unit testing focuses on the smallest unit of the software design (i.e.), the module. The white-box testing techniques were heavily employed for unit testing.

**FUNCTIONAL TESTS:**

Functional test cases involved exercising the code with nominal input values for which the expected results are known, as well as boundary values and special values, such as logically related inputs, files of identical elements, and empty files.

Three types of tests in Functional test:

Ø Performance Test

Ø Stress Test

Ø Structure Test

**PERFORMANCE TEST:**

It determines the amount of execution time spent in various parts of the unit, program throughput, and response time and device utilization by the program unit.

**STRESS TEST:**

Stress Test is those test designed to intentionally break the unit. A Great deal can be learned about the strength and limitations of a program by examining the manner in which a programmer in which a program unit breaks.

**STRUCTURED TEST:**

Structure Tests are concerned with exercising the internal logic of a program and traversing particular execution paths. The way in which White-Box test strategy was employed to ensure that the test cases could Guarantee that all independent paths within a module have been have been exercised at least once.

Ø Exercise all logical decisions on their true or false sides.

Ø Execute all loops at their boundaries and within their operational bounds.

Ø Exercise internal data structures to assure their validity.

Ø Checking attributes for their correctness.

Ø Handling end of file condition, I/O errors, buffer problems and textual errors in output information

# **2. INTEGRATION TESTING:**

Integration testing is a systematic technique for construction the program structure while at the same time conducting tests to uncover errors associated with interfacing. i.e., integration testing is the complete testing of the set of modules which makes up the product. The objective is to take untested modules and build a program structure tester should identify critical modules.

Critical modules should be tested as early as possible. One approach is to wait until all the units have passed testing, and then combine them and then tested. This approach is evolved from unstructured testing of small programs. Another strategy is to construct the product in increments of tested units. A small set of modules are integrated together and tested, to which another module is added and tested in combination. And so on. The advantages of this approach are that, interface dispenses can be easily found and corrected.

The major error that was faced during the project is linking error. When all the modules are combined the link is not set properly with all support files. Then we checked out for interconnection and the links. Errors are localized to the new module and its intercommunications. The product development can be staged, and modules integrated in as they complete unit testing. Testing is completed when the last module is integrated and tested.

**TESTING TECHNIQUES / TESTING STRATRGIES:**

**TESTING:**

Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an as-yet –undiscovered error. A successful test is one that uncovers an as-yet- undiscovered error.

System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently as expected before live operation commences. It verifies that the whole set of programs hang together. System testing requires a test consists of several key activities and steps for run program, string, system and is important in adopting a successful new system. This is the last chance to detect and correct errors before the system is installed for user acceptance testing.

The software testing process commences once the program is created and the documentation and related data structures are designed. Software testing is essential for correcting errors. Otherwise the program or the project is not said to be complete.

Software testing is the critical element of software quality assurance and represents the ultimate the review of specification design and coding.

Testing is the process of executing the program with the intent of finding the error.

A good test case design is one that as a probability of finding an yet undiscovered error.

A successful test is one that uncovers an yet undiscovered error.

Any engineering product can be tested in one of the two ways:

**White box testing:**

This testing is also called as Glass box testing. In this testing, by knowing the specific functions that a product has been design to perform test can be conducted that demonstrate each function is fully operational at the same time searching for errors in each function. It is a test case design method that uses the control structure of the procedural design to derive test cases. Basis path testing is a white box testing.

**Basis path testing:**

Ø Flow graph notation

Ø Cyclomatic complexity

Ø Deriving test cases

Ø Graph matrices Control

**Black Box Testing:**

In this testing by knowing the internal operation of a product, test can be conducted to ensure that “all gears mesh”, that is the internal operation performs according to specification and all internal components have been adequately exercised. It fundamentally focuses on the functional requirements of the software.

The steps involved in black box test case design are:

Ø Graph based testing methods

Ø Equivalence partitioning

Ø Boundary value analysis

Ø Comparison testing

**SOFTWARE TESTING STRATEGIES:**

A software testing strategy provides a road map for the software developer. Testing is a set activity that can be planned in advance and conducted systematically. For this reason a template for software testing a set of steps into which we can place specific test case design methods should be strategy should have the following characteristics:

Ø Testing begins at the module level and works “outward” toward the integration of the entire computer based system.

Ø Different testing techniques are appropriate at different points in time.

Ø The developer of the software and an independent test group conducts testing.

Ø Testing and Debugging are different activities but debugging must be accommodated in any testing strategy.

**INTEGRATION TESTING:**

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with. Individual modules, which are highly prone to interface errors, should not be assumed to work instantly when we put them together. The problem of course, is “putting them together”- interfacing. There may be the chances of data lost across on another’s sub functions, when combined may not produce the desired major function; individually acceptable impression may be magnified to unacceptable levels; global data structures can present problems.

**VALIDATION TESTING:**

Software validation is achieved through a series of tests that demonstrates conformity with requirements. A test plan outlines the classes of test to be conducted and a test procedure defines specific test cases that will be used to demonstrate conformity with requirements. Thus the proposed system under consideration has been tested by validation and found to be working satisfactorily.

**PROGRAM TESTING:**

The logical and syntax errors have been pointed out by program testing. A syntax error is an error in a program statement that in violates one or more rules of the language in which it is written. An improperly defined field dimension or omitted keywords are common syntax error. These errors are shown through error messages generated by the computer. A logic error on the other hand deals with the incorrect data fields, out-off-range items and invalid combinations. Since the compiler s will not deduct logical error, the programmer must examine the output. Condition testing exercises the logical conditions contained in a module. The possible types of elements in a condition include a Boolean operator, Boolean variable, a pair of Boolean parentheses A relational operator or on arithmetic expression. Condition testing method focuses on testing each condition in the program the purpose of condition test is to deduct not only errors in the condition of a program but also other a errors in the program.

**SECURITY TESTING:**

Security testing attempts to verify the protection mechanisms built in to a system well, in fact, protect it from improper penetration. The system security must be tested for invulnerability from frontal attack must also be tested for invulnerability from rear attack. During security, the tester places the role of individual who desires to penetrate system.

**SYSTEM SECURITY MEASURES:**

System security refers to the technical innovations and procedures applied to the hardware and operating systems to protect against deliberate or accidental damage from a defined threat. In contrast, data security is the protection of data from disclosure, modification and destruction.

Data privacy and security are issues that go beyond the scope of system development. They are actually a societal concern. Our organization depends heavily on the use of data bases requires special controls to maintain viable information. These controls are classified in to three general categories.

Ø Physical securities are production from fire flood and other physical damage.

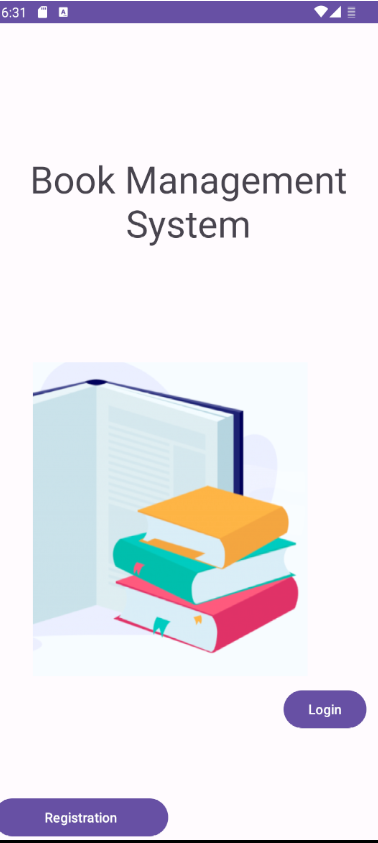
Ø Database integrity through data validation techniques.

Ø Control measures through passwords encryption and monitoring users on a regular basis.

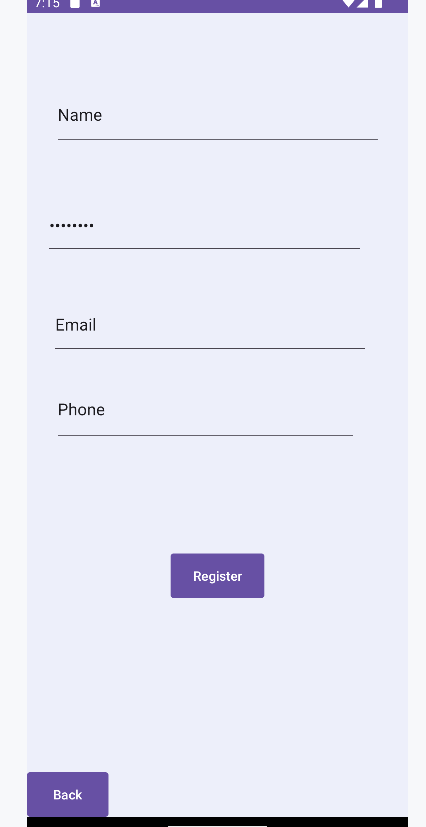
Password is the most common authentication mechanism based on sharing of secret. In a password –based system each user has a password, which may initially be assigned by the system or an administrator. In Online Recruitment System, only administrator is allowed to change the user’s passwords. The system stores all users passwords and users them to authenticate the user. When logging in, the system request the user supplies a presumably secret, user specific password. This way the passwords provide protection to some extent

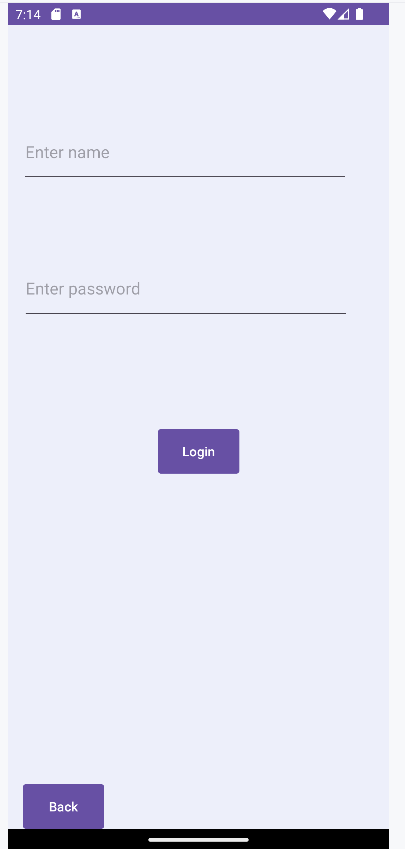
The proper backup of data in the form of hardcopies as well as on paper stationary is a measure of disaster recovery planning.

Sample Screen

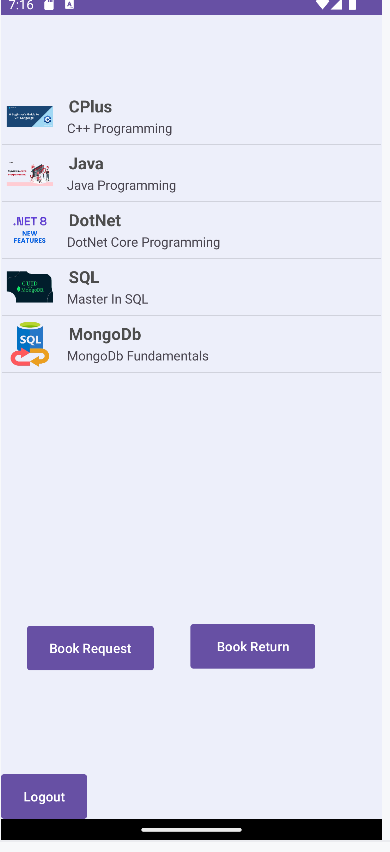


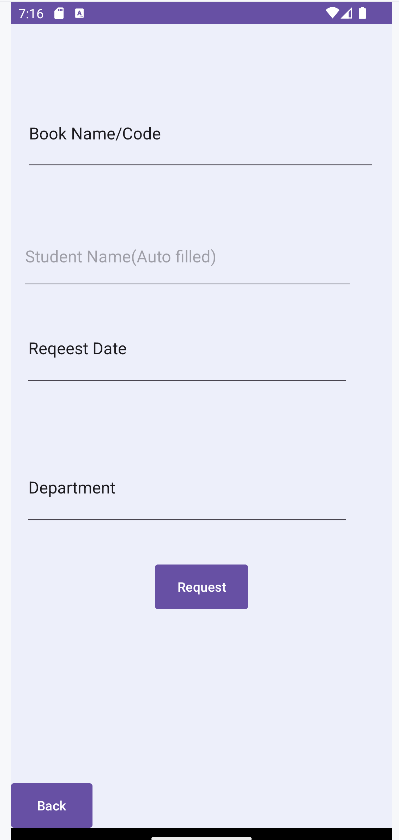
Registration

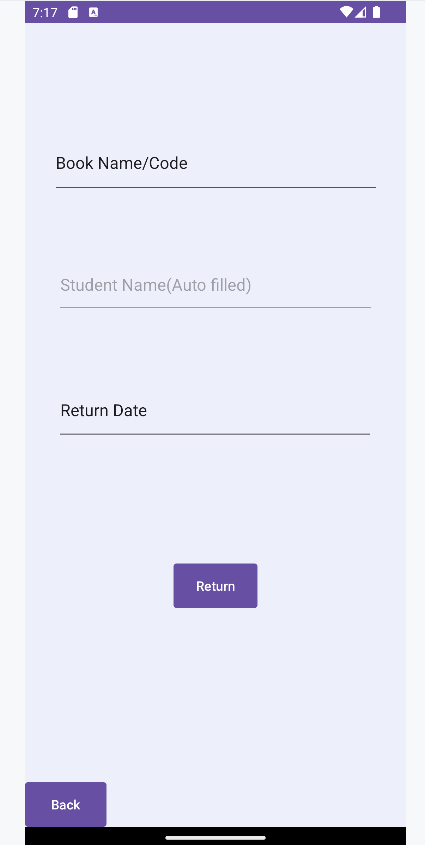




Username : student

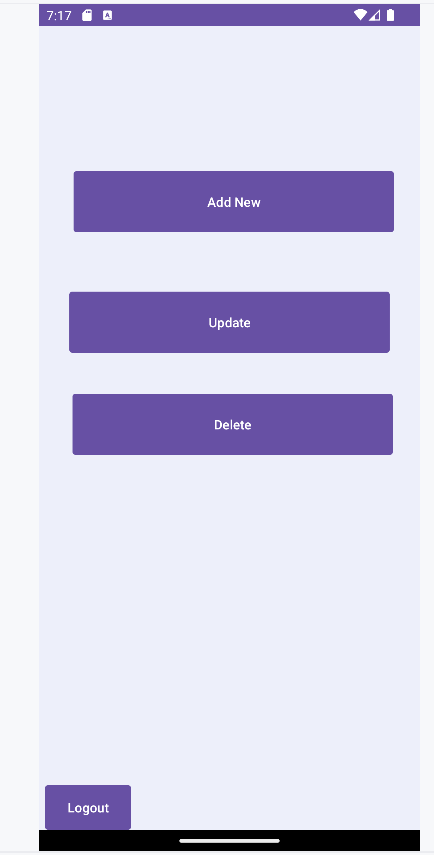
Password : student  
  


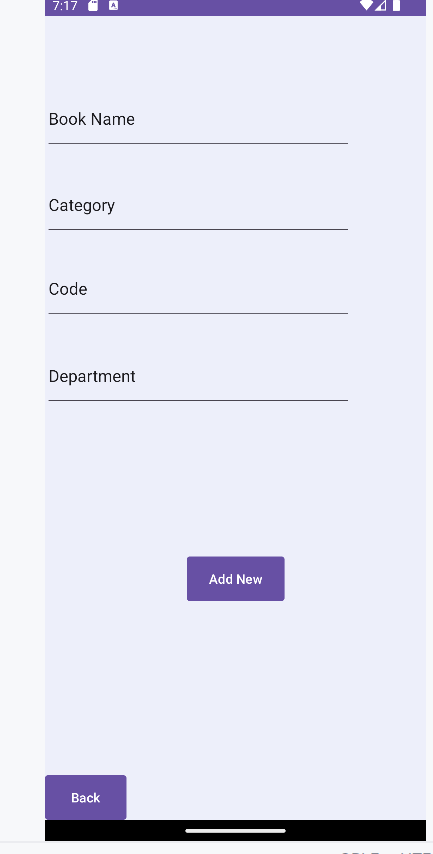


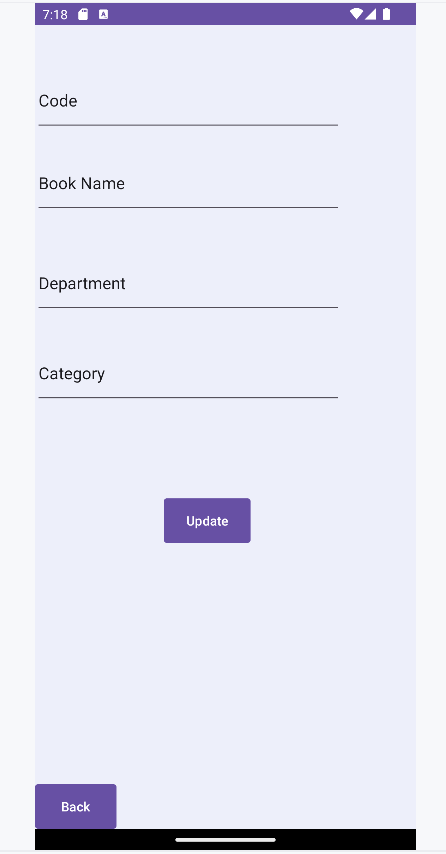
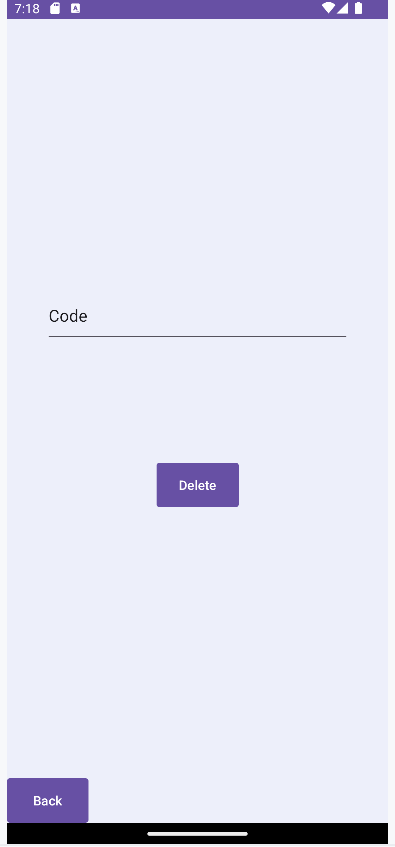


Username : admin

Password : admin





**Conclusion**

This project aims to enhance the conventional approach to book management, addressing the drawbacks associated with manual processes such as sluggishness, inefficiency, decreased security, and challenging administration. Our solution involves the development of an online book management application that streamlines and digitizes the entire workflow. Operating on the Android platform, our application is integrated with a SQLite database.

The frontend components have been meticulously crafted using Android and its associated packages, while the backend is reinforced and linked to the database through Android libraries. As the library workload grows, the flexibility of our application allows for seamless integration of new features, ensuring its continued relevance in the evolving landscape of book management

**References**

1. Pro Android 2 Paperback – Illustrated,

by Dave MacLean (Author), Satya Komatineni (Author), Sayed Hashimi (Author)

2.Android Application Development All-In-One for Dummies, 3ed Paperback

by Barry Burd (Author), John Paul Mueller (Author)

3.MOBILE APPLICATIONS DEVELOPMENT WITH ANDROID

by Meikang Qiu (Author), Wenyun Dai (Author), Keke Gai (Author)

4.Android Programming For Beginners: The Simple Guide to Learning Android Programming Fast! Paperback – Import, 18 December 2019

by Tim Warren (Author)

5.Programmer's Guide to the Brain: With Examples in Python Paperback

by Rob Vermiller (Author)